

**PERAMALAN HARGA EMAS  
MENGUNAKAN *FEEDFORWARD NEURAL NETWORK*  
DENGAN ALGORITMA *BACKPROPAGATION***

**SKRIPSI**

Diajukan Kepada Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Negeri Yogyakarta untuk memenuhi

Sebagian persyaratan guna memperoleh

Gelar Sarjana Sains



Oleh

Felasufah Kusumadewi

10305144034

**PROGRAM STUDI MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS NEGERI YOGYAKARTA**

2014

PERSETUJUAN

Skripsi yang berjudul

**“PERAMALAN HARGA EMAS  
MENGUNAKAN *FEEDFORWARD NEURAL NETWORK*  
DENGAN ALGORITMA *BACKPROPAGATION*”**

Oleh

Felasufah Kusumadewi

NIM. 10305144034

Telah disetujui dan disahkan oleh dosen pembimbing untuk diajukan kepada

Dewan Penguji Skripsi Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Negeri Yogyakarta

Disetujui pada tanggal

22 Mei 2014

Menyetujui,

Dosen Pembimbing



Rosita Kusumawati, M.Sc  
NIP. 198007072005012001

## PENGESAHAN

SKRIPSI DENGAN JUDUL:

**“PERAMALAN HARGA EMAS  
MENGUNAKAN *FEEDFORWARD NEURAL NETWORK*  
DENGAN ALGORITMA *BACKPROPAGATION*”**

Yang disusun oleh:

Nama : Felasufah Kusumadewi

NIM : 10305144034

Prodi : Matematika

Skripsi telah diujikan di depan Dewan Peguji Skripsi pada  
tanggal 30 Mei 2014 dan dinyatakan lulus.

### DEWAN PENGUJI

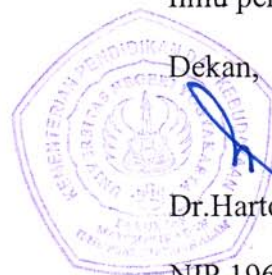
Nama	Jabatan	Tanda Tangan	Tanggal
<u>Rosita Kusumawati, M.Sc</u> NIP.19807072005012001	Ketua Penguji		19 Juni 2014
<u>Eminugroho R.S, M.Sc</u> NIP.198504142009122003	Sekretaris Penguji		19 Juni 2014
<u>Dr. Dhoriva U.W</u> NIP.196603311993032001	Penguji Utama		18 Juni 2014
<u>Kuswari H. M.Kom</u> NIP.197604142005012002	Penguji Pendamping		19 Juni 2014

Yogyakarta, 1 Juli 2014

Fakultas Matematika dan

Ilmu pengetahuan Alam

Dekan,



Dr.Hartono

NIP.196203291987021002

## SURAT PERNYATAAN

Yang bertandatangan dibawah ini saya:

Nama : Felasufah Kusumadewi

NIM : 10305144034

Program Studi : Matematika

Jurusan : Pendidikan Matematika

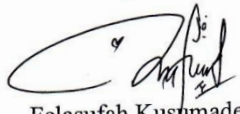
Fakultas : Matematika dan Ilmu Pengetahuan Alam

Judul Skripsi : Peramalan Harga Emas menggunakan *Feedforward Neural Network* dengan Algoritma *Backpropagation*

Menyatakan bahwa skripsi ini benar-benar karya saya sendiri. Sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain kecuali sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim. Apabila ternyata terbukti bahwa pernyataan ini tidak benar maka sepenuhnya menjadi tanggung jawab saya dan saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Yogyakarta, 22 Mei 2014

Yang Menyatakan

  
Felasufah Kusumadewi

NIM. 10305144034

## **MOTTO**

*Sesuatu mungkin mendatangi mereka yang mau menunggu,  
namun hanya didapatkan oleh mereka yang bersemangat  
mengejarinya*

*~ Abraham Loncoln*

## **PERSEMBAHAN**

*Untuk Ibuku tersayang, Setiyo Winarti  
Terimakasih atas kasih sayang yang tiada henti*

*Untuk Bapakku tercinta, Sarwo Atmojo Subroto  
Terimakasih atas bimbingan yang menjadikanku hebat*

*Untuk Adikku terkasih, Fajar Ahmad Suryo Guritno  
Terimakasih untuk canda tawa yang selalu menyemangatiku*

# **PERAMALAN HARGA EMAS MENGGUNAKAN *FEEDFORWARD NEURAL NETWORK* DENGAN ALGORITMA *BACKPROPAGATION***

Oleh:

Felasufah Kusumadewi

NIM. 10305144034

## **ABSTRAK**

*Time series* adalah pengamatan pada suatu variabel dari waktu lampau dan dicatat secara berurutan menurut urutan waktu dengan periode yang tetap. Data dari waktu lampau harga emas merupakan salah satu data *time series*. Pada umumnya orang memilih berinvestasi dalam bentuk emas untuk memperoleh keuntungan. Tujuan dari penelitian ini adalah menjelaskan prosedur pembentukan model *Feedforward Neural Network* (FFNN) dengan Algoritma *Backpropagation* (BP) dan meramalan harga emas menggunakan model tersebut.

Prosedur pembentukan model *Feedforward neural network* dengan algoritma *Backpropagation* pada data *time series* terdiri atas beberapa tahap, yaitu (1) menentukan input berdasarkan plot ACF dan PACF, (2) melakukan pembagian data menjadi 2 yaitu data training dan data testing (3) menormalisasi data, (4) membangun model *Feedforward neural network* dengan algoritma *Backpropagation*, yaitu menentukan jumlah neuron pada lapisan tersembunyi, menentukan input yang optimal, dan menentukan bobot model, (5) denormalisasi dan (6) uji kesesuaian model. Langkah tersebut menghasilkan model yang terbaik, yang dapat digunakan untuk peramalan.

Model FFNN dengan algoritma BP ini diterapkan pada data harga emas bulan Juli 2008 sampai Februari 2014 dengan variabel inputnya yaitu harga emas dunia dan harga minyak dunia. Struktur jaringan terbaik yang diperoleh adalah dengan 3 neuron input dan 9 neuron pada lapis tersembunyi dengan menggunakan fungsi aktivasi *sigmoid bipolar*, fungsi *linear*, dan algoritma *traingdx*. Peramalan harga emas tersebut menghasilkan MAPE 1,8178% pada data *training* dan 5,6808% pada data *testing*. Hasil peramalan untuk bulan Maret 2014 – Mei 2014 adalah US\$ 791,6 per *troy ounce*; US\$ 1188,6 per *troy ounce*; dan US\$ 893,5 per *troy ounce*.

**Kata kunci:** *Feedforward Neural Network*, *Backpropagation*, peramalan, harga emas

## KATA PENGANTAR

Segala puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan nikmat serta hidayah-Nya terutama nikmat kesempatan dan kesehatan sehingga penulis dapat menyelesaikan tugas akhir skripsi ini. Skripsi yang berjudul “Peramalan Harga Emas Menggunakan *Feedforward Neural Network* dengan Algoritma *Backpropagation*” disusun untuk memenuhi salah satu syarat kelulusan guna meraih gelar Sarjana Sains pada Program Studi Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta.

Skripsi ini tidak akan terselesaikan tanpa bantuan, bimbingan, saran dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Dr. Hartono, M.Si selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta dan Bapak Dr. Sugiman, M.Si selaku Ketua Jurusan Pendidikan Matematika Universitas Negeri Yogyakarta.
2. Bapak Dr. Agus Maman Abadi, M.si selaku Ketua Prodi Matematika Universitas Negeri Yogyakarta dan Bapak Nur Hadi Waryanto, M.Eng selaku penasehat akademik
3. Ibu Rosita Kusumawati, M.Sc selaku Dosen Pembimbing yang telah meluangkan waktu untuk memberikan bimbingan dan arahan dalam menyusun skripsi.



4. Seluruh dosen Jurusan Pendidikan Matematika Universitas Negeri Yogyakarta yang telah memberikan ilmu dan pengajaran kepada penulis.
5. Ibu Iko Ramadhani Vianti, ST selaku Division Manager ACT dari PT. Monex Investindo Futures Yogyakarta yang telah memberikan bantuan kepada penulis dalam pengambilan data skripsi.
6. Metza, Dewi, dan Icha atas motivasinya kepada penulis serta kerjasamanya untuk sama-sama berjuang demi cita-cita kita. Serta teman-teman Matswa'10 yang selalu memberikan dukungan kepada penulis.
7. Mbak Tia, Ima, Mbak Cimi, Dek Irma, Dhita, Itna dan teman-teman kos yang selalu mendukung dan membantu penulis dalam menyusun skripsi.
8. Seluruh pihak yang tidak dapat disebutkan satu persatu telah memberikan dukungan, bantuan dan motivasi kepada penulis.

Penulis menyadari adanya ketidakteitian, kekurangan dan kesalahan dalam penulisan tugas akhir skripsi ini. Oleh karena itu, penulis menerima kritik dan saran yang bersifat membangun. Semoga penulisan tugas akhir ini dapat bermanfaat bagi pembaca dan pihak yang terkait.

Yogyakarta, 22 Mei 2014

Penulis

Felasufah Kusumadewi

## DAFTAR ISI

HALAMAN JUDUL .....	i
PERSETUJUAN.....	ii
HALAMAN PENGESAHAN .....	iii
HALAMAN PERNYATAAN.....	iv
MOTTO.....	v
HALAMAN PERSEMBAHAN .....	vi
ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI .....	x
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN.....	xvi
DAFTAR SIMBOL .....	xviii
BAB I PENDAHULUAN .....	1
A. Latar Belakang .....	1
B. Rumusan Masalah .....	4
C. Tujuan Penelitian .....	5

D. Manfaat Penelitian.....	5
E. Metode Penelitian.....	5
<b>BAB II KAJIAN PUSTAKA .....</b>	<b>6</b>
A. Konsep Dasar <i>Time Series</i> .....	6
1. Autokorelasi .....	6
2. Proses <i>White Noise</i> .....	9
3. Pola Data .....	10
B. Konsep Dasar <i>Neural Network</i> .....	12
1. Arsitektur <i>Neural Network</i> .....	14
2. Metode Pelatihan .....	16
3. Fungsi Aktivasi.....	16
C. Investasi Emas.....	18
<b>BAB III PEMBAHASAN .....</b>	<b>22</b>
A. Model <i>Feedforward Neural Network</i> dengan Algoritma <i>Backpropagation</i> .....	22
1. Algoritma Backpropagation.....	23
2. Membangun jaringan <i>feedforward neural network</i> dengan algoritma <i>backpropagation</i> .....	27
B. Penerapan Model <i>Feedforward Neural Network</i> dengan Algoritma <i>Backpropagation</i> untuk Peramalan Harga Emas .....	37
1. Membangun jaringan <i>feedforward neural network</i> dengan algoritma <i>backpropagation</i> untuk peramalan	

harga emas.....	37
2. Peramalan harga emas .....	45
 BAB IV KESIMPULAN DAN SARAN .....	49
A. Kesimpulan .....	49
B. Saran.....	50
 DAFTAR PUSTAKA .....	51
 LAMPIRAN .....	53

## DAFTAR TABEL

Tabel 3. 1 Nilai MSE dan MAPE Hasil Pembelajaran <i>Traindx</i> dengan Algoritma <i>Backpropagation</i> .....	41
Tabel 3. 2 Nilai MSE dan MAPE Input Optimal .....	42

## DAFTAR GAMBAR

Gambar 2.1 Plot autokorelasi pada data <i>time series</i> .....	9
Gambar 2.2 Pola data horisontal.....	11
Gambar 2.3 Pola data musiman.....	11
Gambar 2.4 Pola data siklis.....	12
Gambar 2.5 Pola data Trend.....	12
Gambar 2.6 Arsitektur jaringan <i>neural network</i> sederhana .....	14
Gambar 2.7 Arsitektur jaringan <i>neural network</i> single layer.....	15
Gambar 2.8 Arsitektur jaringan <i>neural network</i> multilayer .....	15
Gambar 2.9 Fungsi aktivasi undak biner ( <i>thershold</i> ) .....	17
Gambar 2.10 Fungsi aktivasi linear (identitas).....	17
Gambar 2.11 Plot harga open emas .....	19
Gambar 3.1 Arsitektur <i>Feedforward Neural Network</i> .....	23
Gambar 3.2 Plot ACF (kiri) dan PACF (kanan) data harga open emas periode bulanan mulai dari bulan Juli 2008 sampai Februari 2014 .....	38
Gambar 3.3 Plot ACF (kiri) dan PACF (kanan) data harga open minyak dunia periode bulanan mulai dari bulan Juli 2008 sampai Februari 2014.....	38
Gambar 3.4 Plot ACF model FFNN algoritma <i>backpropagation</i> dengan 9 neuron pada lapis tersembunyi dan $(x_1, x_2, x_3)$ sebagai input .....	44
Gambar 3.5 Plot PACF model FFNN algoritma <i>backpropagation</i> dengan 9 neuron pada lapis tersembunyi dan $(x_1, x_2, x_3)$ sebagai input .....	44

Gambar 3.6 Arsitektur model <i>Feedforward Neural Network</i> dengan	
algoritma <i>backpropagation</i> pada peramalan harga open emas .....	45

## DAFTAR LAMPIRAN

Lampiran I Data harga open emas dunia dan harga open minyak dunia(Juli 2008 - Februari 2014) .....	53
Lampiran II Data <i>Training</i> .....	55
Lampiran III Data <i>Testing</i> .....	57
Lampiran IV Hasil normalisasi data .....	58
Lampiran V Program <i>Feedforward neural network</i> dengan algoritma <i>Backpopagation</i> menggunakan MATLAB ( dengan fungsi aktivasi <i>sigmoid bipolar</i> pada hidden layer, fungsi <i>linear</i> pada lapisan output dan menggunakan <i>traindx</i> ).....	60
Lampiran VI Program <i>Feedforward neural network</i> dengan algoritma <i>Backpopagation</i> untuk mencari input optimum dengan mengeliminasi $x_1$ menggunakan MATLAB ( dengan fungsi aktivasi <i>sigmoid bipolar</i> pada hidden layer dengan 9 neuron, fungsi <i>linear</i> pada lapisan output dan menggunakan pembelajaran <i>traindx</i> ).....	63
Lampiran VII Program <i>Feedforward neural network</i> dengan algoritma <i>Backpopagation</i> untuk mencari input optimum dengan mengeliminasi $x_2$ menggunakan MATLAB ( dengan fungsi aktivasi <i>sigmoid bipolar</i> pada hidden layer dengan 9 neuron, fungsi <i>linear</i> pada lapisan output dan menggunakan pembelajaran <i>traingdx</i> ).....	66
Lampiran VIII Program <i>Feedforward neural network</i> dengan algoritma <i>Backpopagation</i> untuk mencari input optimum dengan mengeliminasi $x_3$ menggunakan MATLAB ( dengan fungsi aktivasi <i>sigmoid bipolar</i> pada hidden	



layer dengan 9 neuron, fungsi <i>linear</i> pada lapisan output dan menggunakan pembelajaran <i>traingdx</i> ).....	69
Lampiran IX Hasil pembobotan.....	72
Lampiran X Program <i>Feedforward neural network</i> dengan algoritma <i>Backpopagation</i> untuk peramalan harga open emas menggunakan MATLAB (dengan fungsi aktivasi <i>sigmoid bipolar</i> pada hidden layer, fungsi <i>linear</i> pada lapisan output dan menggunakan pembelajaran <i>traingdx</i> dengan 9 neuron tersembunyi dan $x_1, x_2, x_3$ sebagai input).....	74

## DAFTAR SIMBOL

$n$  = banyak data pengamatan

$\mu$  = rata-rata

$\gamma_k$  = autokovarian pada *lag*  $k$

$\rho_k$  = autokorelasi pada *lag*  $k$

$t$  = waktu pengamatan,  $t = 1, 2, 3, \dots$

$X_t$  = pengamatan pada waktu  $t$

$X_{t+k}$  = pengamatan pada waktu  $t+k$

$X_{t-k}$  = nilai pengamatan pada *lag*  $k$

$r_k$  = koefisien autokorelasi untuk *lag*  $k$

$\bar{Y}$  = nilai rata-rata variabel  $Y$

$SE(r_k)$  = standar *error* koefisien korelasi pada *lag*  $k$

$x_i$  = lapisan input

$v_{ji}$  = bobot yang menghubungkan neuron *ke-i* pada lapisan input ke neuron *ke-j* pada lapisan tersembunyi

$v_{j0}$  = bobot yang menghubungkan bias ke neuron *ke-j* pada lapisan tersembunyi

$z_j$  = neuron pada lapisan tersembunyi

$w_{kj}$  = bobot yang menghubungkan neuron  $ke-j$  pada lapisan tersembunyi ke neuron  $ke-k$  pada lapisan output

$w_{k0}$  = bobot yang menghubungkan bias pada neuron  $ke-k$  pada lapisan tersembunyi ke lapisan output

$\delta_k$  = unit *error* yang akan dipakai dalam perubahan bobot lapis dibawahnya

$\Delta w_{kj}$  = perubahan bobot  $w_{kj}$

$\Delta v_{ji}$  = perubahan bobot  $v_{ji}$

$y_k$  = output jaringan

$s^2$  = varians

## **BAB I**

### **PENDAHULUAN**

#### **A. Latar Belakang**

Investasi pada hakikatnya merupakan komitmen terhadap sejumlah sumber daya pada saat ini dengan tujuan untuk mendapatkan keuntungan di masa depan (Abdul halim, 2005 : 4). Umumnya investasi dibedakan menjadi dua, yaitu investasi pada aset-aset finansial (*financial assets*) dan investasi pada aset-aset riil (*real assets*). Investasi pada aset-aset finansial dilakukan di pasar uang, misalnya berupa sertifikat deposito, *commercial paper*, surat berharga pasar uang, dan lain-lain. Investasi juga dapat dilakukan di pasar modal, misalnya berupa saham, obligasi, waran, opsi, dan lain-lain. Sedangkan investasi pada aset-aset riil dapat berbentuk pembelian aset produktif, pendirian pabrik, pembukaan pertambangan, pembukaan perkebunan dan lainnya.

Emas merupakan salah satu logam mulia yang bernilai di dunia yang bersifat lunak, tahan korosi, mudah ditempa. Para pakar investasi seringkali menganjurkan untuk berinvestasi pada emas karena emas merupakan sarana lindung nilai klasik untuk melawan inflasi dan menambah nilai dalam kondisi ketidakstabilan fluktuasi nilai mata uang (Maya Apriyanti, 2012 : 3). Investasi dalam emas dibedakan menjadi dua jenis yaitu, investasi pada saham emas dan investasi pada emas batangan. Harga saham open emas dari tahun 2008 sampai dengan 2014 nyaris tidak pernah turun tajam. Data yang diperoleh dari PT. Monex Investindo Futures yaitu pada bulan Juli 2008 harga emas US\$ 914,40 per *troy ounce* dan pada Januari 2014 harga emas mencapai US\$ 1245,05 per

*troy ounce*. Begitu juga dengan harga emas batangan yang menyesuaikan harga saham emas. Harga emas yang nyaris tidak pernah turun tersebut melatarbelakangi minat beli investor terhadap emas.

Selain itu, faktor yang melatarbelakangi minat beli investor terhadap emas adalah pembelian emas yang melonjak tajam oleh Negara China dan India. Ancaman inflasi masa depan akibat kebijakan cetak uang oleh bank sentral negara-negara maju (kebijakan ini berakibat pada kehancuran nilai tukar mata uang global) termasuk salah satu faktor yang mempengaruhi kenaikan atau penurunan harga emas. Faktor yang mempengaruhi kenaikan atau penurunan harga emas yang lain adalah (1) krisis financial, (2) naiknya permintaan emas di pasaran, (3) kurs dollar, (4) harga minyak, dan (6) situasi politik dunia.

Agar tujuan investasi tercapai, maka sebelum memasuki dunia investasi diperlukan pengetahuan keuntungan dan risiko yang didapat ketika terjun di bidang investasi. Harapan keuntungan dalam dunia investasi sering juga disebut sebagai *return*. Risiko investasi bisa diartikan sebagai kemungkinan terjadinya perbedaan antara *return aktual* dengan *return* yang diharapkan. Dua konsep ini, risiko maupun *return*, bagaikan dua sisi mata uang yang selalu berdampingan. Artinya, dalam berinvestasi di samping menghitung *return* yang diharapkan investor juga harus memperhatikan risiko yang ditanggung (Abdul Halim, 2005 : 5). Pengetahuan ini penting sebagai pegangan ketika memasuki dunia investasi yang penuh risiko dan ketidakpastian.

Salah satu pengetahuan penting dalam berinvestasi emas adalah peramalan harganya. Peramalan harga emas diperlukan bagi investor untuk mengetahui kecenderungan harga emas di masa datang. Peramalan adalah proses perkiraan (pengukuran) besarnya atau jumlah sesuatu pada waktu yang akan datang berdasarkan data pada masa lampau (*time series*) yang dianalisis secara ilmiah khususnya menggunakan metode statistika (Sudjana, 1996 : 254). Peramalan harga emas bertujuan untuk mengetahui peluang investasi harga emas di masa yang akan datang sehingga dapat digunakan sebagai pertimbangan oleh investor emas untuk mengetahui perubahan harga emas.

Metode peramalan sangat banyak dan seringkali memerlukan asumsi-asumsi yang harus dipenuhi, namun terdapat juga model yang tidak memerlukan asumsi-asumsi salah satunya **Neural Network (NN)**. Model NN dibedakan menjadi 2, yaitu *Feedforward Neural Network (FFNN)*, dimana proses pelatihan berjalan maju dari lapisan input menuju lapisan output selanjutnya, *Recurrent Neural Network (RNN)* yang proses pembelajarannya paling sedikit ada satu koneksi umpan balik supaya terjadi proses siklis (Fausset, 1994 : 12). Yang termasuk dalam kelas FFNN adalah *Backpropagation Neural Network*, *Radial Basis Function Network*, *General Regression Neural Network*. Algoritma *Backpropagation* dalam FFNN merupakan model yang sederhana jika digunakan untuk menyelesaikan masalah data *time series*. Salah satu data *time series* adalah harga emas.

Penggunaan FFNN untuk analisis data *time series* secara luas telah banyak dilakukan, antara lain Edy Supriyanto (2004) menerapkan jaringan

syaraf tiruan untuk memprediksi harga saham, Luh Putu Widya (2012) menggunakan GRNN pada peramalan data *time series*, Sielvy Evatiana (2013) mengaplikasikan model *neuro fuzzy* untuk memprediksi harga emas, Dian Tri Handayani dan Agus Maman Abadi (2012) menggunakan model *neuro fuzzy* untuk peramalan nilai tukar rupiah terhadap yen, Desty Anna Kumalasari (2013) mengkasifikasikan metode jaringan syaraf tiruan (*Artificial Neural Networks*) dengan fungsi *radial basis* pada risiko kredit, Widatul Milla (2012) menerapkan model *neural networks* dengan algoritma *recurrent* sebagai metode peramalan harga koin emas di pegadaian, Setyawati (2013) mengaplikasikan model *Elman recurrent neural networks* dengan algoritma *backpropagation* untuk peramalan Indeks Harga Konsumen di Yogyakarta, Ambar Sulistyaningsih (2013) mengaplikasikan model *recurrent neural networks* dengan variasi kalender islam pada data kunjungan wisata candi Prambanan, Rika Wahyuni (2005) menggunakan NN dan model ARIMA untuk peramalan finansial. Oleh karena itu dalam penelitian ini akan dikaji pemodelan *Neural Network* dengan Algoritma *Backpropagation* untuk data *time series* dan aplikasinya pada data harga emas.

## **B. Rumusan Masalah**

Berdasarkan latar belakang, maka dalam skripsi ini penulis merumuskan masalah yaitu :

1. Bagaimana prosedur pembentukan model FFNN menggunakan algoritma *Backpropagation* ?

2. Bagaimana hasil peramalan harga emas dengan model FFNN menggunakan algoritma *Backpropagation* ?

### **C. Tujuan Penelitian**

Sesuai dengan pokok permasalahan yang telah dirumuskan, maka tujuan penulisan tugas akhir ini adalah

1. Menjelaskan prosedur pembentukan model FFNN dengan algoritma *Backpropagation* pada data *time series*.
2. Meramalkan harga emas menggunakan model FFNN dengan algoritma *Backpropagation*.

### **D. Manfaat Penelitian**

Adapun manfaat yang diperoleh dari penulisan tugas akhir ini adalah:

1. Sebagai referensi atau bahan acuan untuk mengaplikasikan model FFNN dengan algoritma *Backpropagation* dalam melakukan peramalan serta sebagai bahan informasi untuk penelitian selanjutnya.
2. Bagi para investor, diharapkan penelitian ini dapat menjadi bahan pertimbangan untuk melakukan analisis investasi emas.



## BAB II

### KAJIAN PUSTAKA

#### A. Konsep Dasar Time series

*Time series* adalah pengamatan pada suatu variabel dari waktu lampau dan dicatat secara berurutan menurut urutan waktu dengan periode yang tetap (Hanke & Wichern, 2004 : 58). Pada umumnya pencatatan ini dilakukan dalam periode tertentu misalnya harian, bulanan, tahunan dan sebagainya, sedangkan analisis *time series* adalah suatu metode kuantitatif untuk menentukan pola data masa lampau yang telah dikumpulkan secara teratur. Jika telah menemukan pola data tersebut, maka dapat digunakan untuk peramalan di masa mendatang. Beberapa konsep dasar dalam analisis *time series* adalah autokorelasi, konsep *white noise* dan plot data.

##### 1. Autokorelasi

Autokorelasi merupakan suatu korelasi pada data *time series* antara  $X_t$  dengan  $X_{t+k}$ . Untuk mendefinisikan autokorelasi diperlukan definisi autokovarians. Autokovarians dan autokorelasi antara  $X_t$  dan  $X_{t+k}$  berturut-turut dapat didefinisikan sebagai berikut (Wei, 2006 : 10):

$$\gamma_k = Cov(X_t, X_{t+k}) = E[(X_t - \mu)(X_{t+k} - \mu)] \quad (2.1)$$

dan

$$\rho_k = r_k = \frac{Cov(X_t, X_{t+k})}{\sqrt{Var(X_t)}\sqrt{Var(X_{t+k})}} = \frac{\gamma_k}{\gamma_0} \quad (2.2)$$

dengan :

$X_t$  = pengamatan pada waktu ke- $t$

$X_{t+k}$  = pengamatan pada waktu ke-t+k

Dimana  $Var(X_t) = Var(X_{t+k}) = \gamma_0$  dan  $\rho_0 = 1$ . Sebagai fungsi dari k,  $\gamma_k$  disebut fungsi autokovariansi dan  $\rho_k$  disebut sebagai fungsi autokorelasi (*autocorrelation function*) atau ACF, yang mewakili kovarians dan korelasi antara  $X_t$  dan  $X_{t+k}$  dari proses yang sama, hanya dipisahkan oleh *time lag-k*. Jika  $X_t$  dan  $X_{t+k}$  independen maka  $\gamma_k = Cov(X_t, X_{t+k}) = 0$  tetapi tidak berlaku sebaliknya.  $X_t$  dan  $X_{t+k}$  dikatakan tidak berkorelasi jika  $\rho_k = 0$ . Dua variabel dengan hubungan negatif sempurna memiliki koefisien korelasi sebesar -1. Di lain kasus, dua variabel dengan hubungan positif sempurna memiliki koefisien korelasi sebesar +1. Dengan demikian, koefisien korelasi bervariasi antara -1 dan +1 (Hanke&Winchern, 2004 : 35).

Hipotesis untuk menguji signifikansi autokorelasi dirumuskan sebagai

$H_0: \rho_k = 0$  (autokorelasi pada lag ke-k tidak signifikan)

$H_1: \rho_k \neq 0$  (autokorelasi pada lag ke k signifikan)

Uji signifikansi menggunakan distribusi t, dengan statistik uji

$$t = \frac{r_k}{SE(r_k)} \quad (2.3)$$

Standar *error* dari koefisien autokorelasi menggunakan rumus sebagai berikut

(Hanke&Wichern, 2004 : 64):

$$SE(r_k) = \sqrt{\frac{1+2 \sum_{i=1}^{k-1} r_i^2}{n}} \quad (2.4)$$

dengan

$SE(r_k)$ : standar *error* koefisien korelasi pada lag k

$r_k$  : koefisien korelasi pada lag k

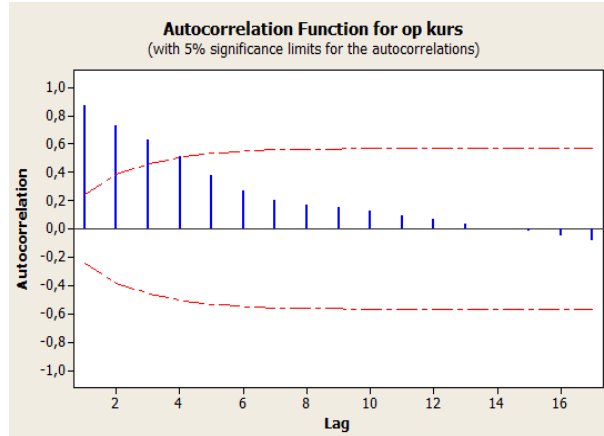
n : banyaknya pengamatan

Koefisien autokorelasi pada lag k dikatakan signifikan jika  $t_{hitung} > t_{(n-1)}\left(\frac{\alpha}{2}\right)$  atau  $t_{hitung} < -t_{(n-1)}\left(\frac{\alpha}{2}\right)$  dan apabila menggunakan nilai Sig. maka koefisien autokorelasi pada lag k dikatakan signifikan jika  $\text{Sig.} < \frac{\alpha}{2}$ .

Signifikansi koefisien autokorelasi juga dapat dilihat dari selang kepercayaan dengan pusat  $r_k = 0$  yang apabila dilihat dari tampilan plot fungsi berupa garis putus-putus yang berwarna merah. Selang kepercayaan tersebut dicari dengan cara menghitung

$$0 \pm t_{(n-1)}\left(\frac{\alpha}{2}\right) \times SE(r_k) \quad (2.5)$$

Pada gambar 2.1 menunjukkan plot autokorelasi pada deret berkala dimana garis yang keluar dari batas signifikan menunjukkan autokorelasi signifikan.



Gambar 2.1 Plot autokorelasi pada data time series  
Autokorelasi parsial digunakan untuk mengukur tingkat keeratan antara  $X_t$  dan  $X_{t-k}$ , apabila pengaruh dari *time* lag 1, 2, 3,..., dan seterusnya sampai  $k-1$  dihilangkan. Autokorelasi parsial ditentukan dengan rumus sebagai berikut

$$\phi_{kk} = \frac{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{k-2} & \rho_1 \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{k-3} & \rho_2 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \dots & \rho_1 & \rho_k \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{k-2} & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{k-3} & \rho_{k-2} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \dots & \rho_1 & 1 \end{vmatrix}} \quad (2.6)$$

Dengan  $\phi_{kk}$  adalah autokorelasi parsial antara  $X_t$  dan  $X_{t+k}$ . Karena merupakan fungsi dari  $k$ , himpunan  $\{\phi_{kk}; k = 0, 1, 2, \dots\}$  dinamakan fungsi autokorelasi parsial (*partial autocorrelation function*), disingkat dengan PACF (Wei, 2006: 15).

## 2. Proses white noise

Sebuah proses  $\{e_t\}$  disebut *white noise* jika merupakan serangkaian variabel acak yang tidak berkorelasi dan berdistribusi tertentu dengan rata-rata

tetap  $E(e_t) = \mu$  biasanya bernilai 0, variansi konstanta  $Var(e_t) = \sigma^2$  dan  $Cov(e_t, e_{t+k}) = 0$  untuk semua  $k \neq 0$  (Wei, 2006 : 16). Dengan demikian proses dari *white noise*  $\{e_t\}$  adalah stasioner dengan fungsi autokovariansi:

$$\gamma_k = \begin{cases} \sigma_a^2 & k = 0, \\ 0 & k \neq 0, \end{cases} \quad (2.7)$$

fungsi autokorelasi

$$\rho_k = \begin{cases} 1 & k = 0, \\ 0 & k \neq 0, \end{cases} \quad (2.8)$$

dan fungsi autokorelasi parsial

$$\phi_{kk} = \begin{cases} 1 & k = 0, \\ 0 & k \neq 0, \end{cases} \quad (2.9)$$

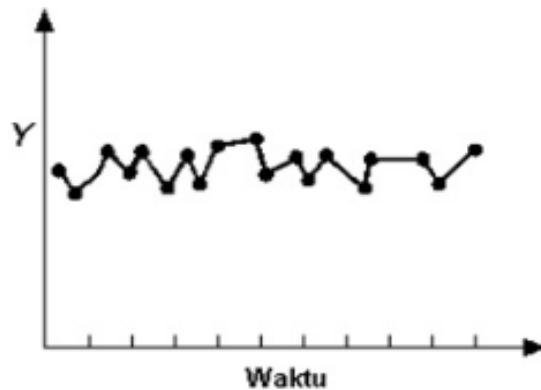
karena menurut definisi  $\rho_0 = \phi_{00} = 1$  untuk semua proses, jika dalam autokorelasi dan autokorelasi parsial hanya mengacu pada  $\rho_0$  dan  $\phi_{kk}$  untuk  $k \neq 0$ . Konsep dasar dari proses *white noise* adalah bahwa ACF dan PACF sama dengan nol.

### 3. Pola Data

Langkah penting dalam memilih suatu metode deret berkala (*time series*) yang tepat adalah mempertimbangkan jenis pola data, sehingga metode yang paling tepat dengan pola tersebut dapat diuji. Pola data dapat dibedakan menjadi empat jenis (Makridakis, Wheelwright, & McGee, 1999 : 10) , yaitu :

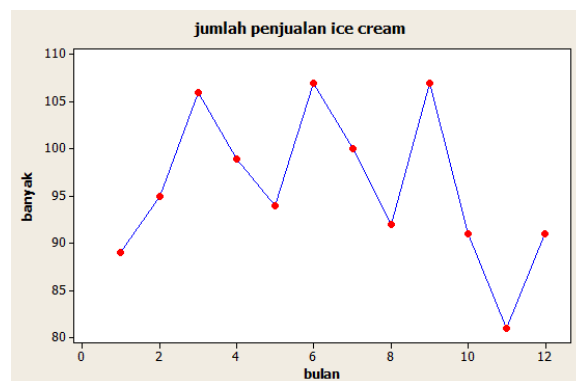
- a. Pola horisontal terjadi bilamana nilai data berfluktuasi di sekitar nilai rata-rata yang konstan. Suatu produk yang penjualannya tidak meningkat atau menurun

selama waktu tertentu termasuk jenis deret ini. Gambar 2.2 menunjukkan suatu pola khas dari data horisontal.



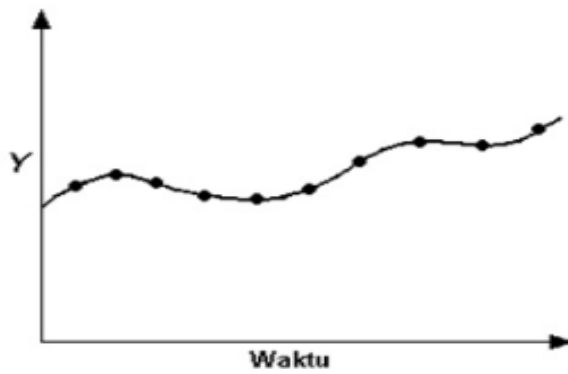
Gambar 2.2 Pola Data Horisontal

- b. Pola musiman terjadi bilamana suatu deret dipengaruhi oleh faktor musiman (misalnya kuartal tahun tertentu, bulanan, atau hari-hari pada minggu tertentu). Penjualan dari produk seperti minuman ringan dan eskrim menunjukkan jenis pola data ini. Gambar 2.3 menunjukkan pola musiman.



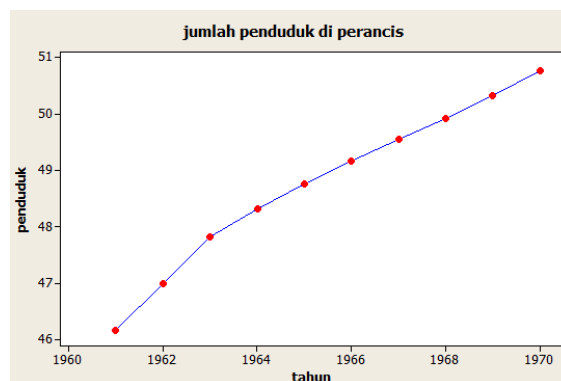
Gambar 2.3 Pola Data Musiman

- c. Pola siklis terjadi bilamana datanya dipengaruhi oleh fluktuasi ekonomi jangka panjang seperti yang berhubungan dengan siklus bisnis. Penjualan produk mobil, baja dan peralatan utama lainnya menunjukkan jenis pola data ini seperti ditunjukkan pada gambar 2.4.



Gambar 2.4 Pola Data Siklis

- d. Pola trend terjadi bilamana terdapat kenaikan atau penurunan jangka panjang dalam data. Penjualan banyak perusahaan, produk buruto nasional (GNP) dan berbagai indikator bisnis ekonomi lainnya mengikuti suatu pola trend selama perubahannya sepanjang waktu. Gambar 2.5 menunjukkan salah satu pola trend seperti itu.



Gambar 2.5 Pola Data Trend

## B. Kosep Dasar *Neural Networks*

*Neural network* adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi (Jong, 2005 : 2). *Neural network* telah diaplikasikan dalam berbagai bidang diantaranya *pattern recognition*, *medical diagnostic*, *signal processing*, dan peramalan. Meskipun

banyak aplikasi menjanjikan yang dapat dilakukan oleh neural network, namun neural network juga memiliki beberapa keterbatasan umum, yaitu ketidakakuratan hasil yang diperoleh. *Neural network* bekerja berdasarkan pola yang terbentuk pada inputnya.

*Neural network* terdiri atas elemen-elemen untuk pemrosesan informasi yang disebut dengan *neuron*, unit, sel atau *node* (Fausset , 1994 : 3). Setiap *neuron* dihubungkan dengan *neuron* lainnya dengan suatu *connection link*, yang direpresentasikan dengan *weight*/bobot. Metode untuk menentukan nilai *weight* disebut dengan *training*, *learning*, atau algoritma. Setiap *neuron* menggunakan fungsi aktivasi pada net input untuk menentukan prediksi *output*.

*Neuron-neuron* dalam *neural network* disusun dalam grup, yang disebut dengan *layer* (lapis) . Susunan *neuron-neuron* dalam lapis dan pola koneksi di dalam dan antarlapis disebut dengan arsitektur jaringan. Arsitektur ini merupakan salah satu karakteristik penting yang membedakan *neural network*. Secara umum ada tiga lapis yang membentuk *neural network*:

#### 1) Lapis input

Unit-unit di lapisan input disebut unit-unit input. Unit-unit input tersebut menerima pola inputan dari luar yang menggambarkan suatu permasalahan. banyak *node* atau neuron dalam lapis input tergantung pada banyaknya input dalam model dan setiap input menentukan satu neuron.

#### 2) Lapis tersembunyi (*hidden layer*)

Unit-unit dalam lapisan tersembunyi disebut unit-unit tersembunyi, di mana outputnya tidak dapat diamati secara langsung. Lapis tersembunyi terletak



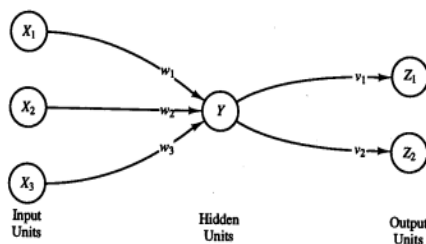
di antara lapis input dan lapis output, yang dapat terdiri atas beberapa lapis tersembunyi.

### 3) Lapis output

Unit-unit dalam lapisan output disebut unit-unit output. Output dari lapisan ini merupakan solusi *Neural Network* terhadap suatu permasalahan. Setelah melalui proses *training*, *network* merespon input baru untuk menghasilkan output yang merupakan hasil peramalan.

## 1. Arsitektur *Neural Network*

Pengaturan *neuron* ke dalam lapisan, pola hubungan dalam lapisan, dan di antara lapisan disebut arsitektur *neural network* (Fausset, 1994 : 12). Arsitektur jaringan *neural network* diilustrasikan dalam gambar 2.6 yang terdiri dari unit input, unit output, dan satu unit tersembunyi. *Neural network* sering diklasifikasikan sebagai *single layer* dan *multilayer*.

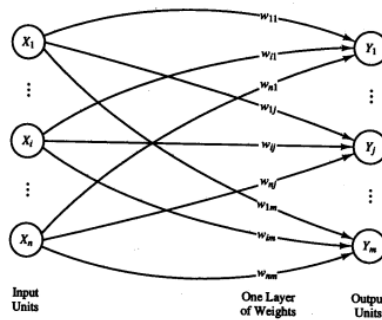


Gambar 2.6 Arsitektur jaringan *neural network* sederhana

### a. *Single layer*

Sebuah jaringan *single layer* memiliki satu lapisan bobot koneksi (Fausset, 1994 : 12). Ciri khas dari *single layer* terlihat dalam gambar 2.7, dimana unit

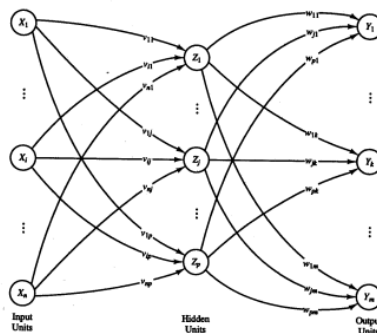
input yang menerima sinyal dari dunia luar terhubung ke unit output tetapi tidak terhubung ke unit input lain, dan unit-unit output yang terhubung ke unit output lainnya.



Gambar 2.7 Arsitektur jaringan *neural network single layer*

#### b. Multilayer

Jaring *multilayer* adalah jaringan dengan satu atau lebih lapisan tersembunyi antara unit input dan unit output (Fausset, 1994 : 14). Biasanya, ada lapisan bobot antara dua tingkat yang berdekatan unit (input, tersembunyi, atau output). Jaringan *multilayer* yang di ilustrasikan pada gambar 2.8 memecahkan masalah yang lebih rumit daripada jaring *single layer*, dan juga pelatihannya mungkin lebih sulit.



Gambar 2.8 Arsitektur jaringan *neural network multilayer*

## 2. Metode Pelatihan

Selain arsitektur, metode pengaturan nilai bobot (training) merupakan karakteristik yang penting dalam jaringan *Neural Network* (Fausset, 1994 : 15). Metode pelatihan pada *Neural network* dibagi menjadi dua jenis, yaitu :

### a. Pelatihan Terawasi

Pelatihan ini dilakukan dengan adanya urutan vektor pelatihan, atau pola yang masing-masing terkait dengan vektor target output. Bobot kemudian disesuaikan untuk algoritma pembelajaran. Proses ini dikenal sebagai pelatihan terawasi.

### b. Pelatihan tak Terawasi

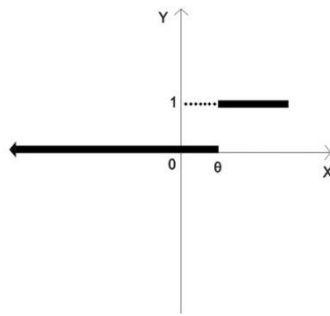
Pada pelatihan ini jaring saraf mengatur segala kinerja dirinya sendiri, mulai dari masukan vektor hingga menggunakan data training untuk melakukan pembelajaran.

## 3. Fungsi Aktivasi

Fungsi aktivasi yang akan menentukan apakah sinyal dari input neuron akan diteruskan atau tidak (Jong, 2005 : 23). Ada beberapa fungsi aktivasi yang sering digunakan dalam *Neural Network*, antara lain:

### a. Fungsi undak biner (*Threshold*)

Fungsi undak biner dengan menggunakan nilai ambang sering juga disebut dengan fungsi nilai ambang (*Threshold*) atau fungsi Heaviside.



Gambar 2.9. fungsi aktivasi undak biner (*threshold*)

Fungsi undak biner (dengan nilai ambang  $\theta$ ) dirumuskan sebagai

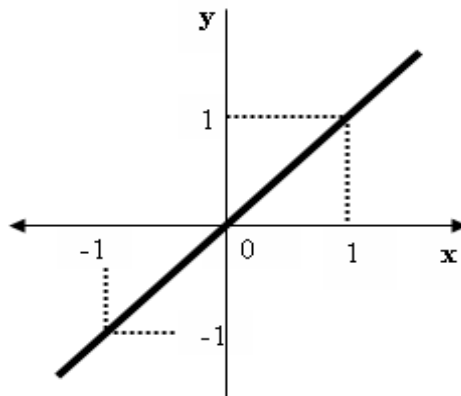
$$y = \begin{cases} 0, & \text{jika } x < \theta \\ 1, & \text{jika } x \geq \theta \end{cases} \quad (2.10)$$

#### b. Fungsi Linear (Identitas)

Fungsi linear memiliki nilai output yang sama dengan nilai inputnya.

Fungsi ini dirumuskan sebagai :

$$y = x \quad (2.11)$$



Gambar 2.10. fungsi aktivasi linear (identitas)

#### c. Fungsi *Sigmoid Biner*

Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *Backpropagation*. Fungsi *sigmoid biner* memiliki nilai pada

range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai output yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan syaraf yang nilai outputnya 0 atau 1. Fungsi *sigmoid biner* dirumuskan sebagai :

$$y = f(x) = \frac{1}{1+e^{-x}} \quad (2.12)$$

$$\text{dengan : } f'(x) = f(x)[1 - f(x)] \quad (2.13)$$

#### d. Fungsi *Sigmoid Bipolar*

Fungsi *sigmoid bipolar* hampir sama dengan fungsi *sigmoid biner*, hanya saja output dari fungsi ini memiliki range antara 1 sampai -1. Fungsi *sigmoid bipolar* dirumuskan sebagai :

$$y = f(x) = \frac{1-e^{-x}}{1+e^{-x}} \quad (2.14)$$

$$\text{dengan: } f'(x) = \frac{1}{2} [1 + f(x)][1 - f(x)] \quad (2.15)$$

Fungsi ini sangat dekat dengan fungsi *hyperbolic tangent*. Keduanya memiliki range antara -1 sampai 1. Untuk fungsi *hyperbolic tangent*, dirumuskan sebagai :

$$y = f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.16)$$

Atau :

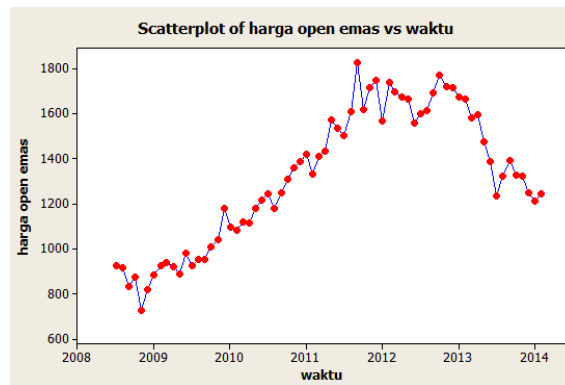
$$y = f(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \quad (2.17)$$

$$\text{dengan : } f'(x) = [1 + f(x)][1 - f(x)] \quad (2.18)$$

### C. Investasi Emas

Pada umumnya orang memilih berinvestasi dalam bentuk emas untuk memperoleh keuntungan (Maya Apriyanti, 2012 : 37). Investasi emas dibedakan

menjadi 2 yaitu, emas batangan dan saham emas (sertifikat). Keduanya mempunyai peluang investasi dan resiko masing-masing. Berinvestasi dalam emas batangan membutuhkan biaya untuk menyewa *safe deposit box* dan memungkinkan resiko lebih besar daripada berinvestasi emas dalam bentuk saham. Dalam berinvestasi emas berbentuk saham, investor hanya perlu keahlian membaca bursa saham. Dalam tulisan ini, penulis meramalkan harga saham open emas yang diperoleh dari PT. Monex Investindo Futures.



Gambar 2.11. plot harga open emas

Emas memiliki *supply* yang terbatas dan tidak mudah didapat, sementara permintaan terhadap emas tidak pernah berkurang, akibatnya harga emas cenderung mengalami kenaikan dari tahun ke tahun seperti yang terlihat pada gambar 2.11. Pada kenyataan sehari-hari, harga emas tidak hanya bergantung kepada situasi permintaan dan penawaran. Harga emas juga dipengaruhi oleh situasi perekonomian secara keseluruhan. Faktor-faktor yang mempengaruhi harga emas sebagai berikut (Maya Apriyanti, 2012 : 58):

## 1. Inflasi

Inflasi terjadi jika nilai mata uang mengalami penurunan sehingga harga barang di pasaran mengalami kenaikan. Hal ini didorong oleh meningkatnya permintaan barang dan jasa yang kemudian diperparah dengan tersendatnya distribusi barang. Kondisi inflasi seperti inilah yang mendorong naiknya harga emas.

## 2. Krisis finansial

Saat terjadi krisis finansial, orang lebih jeli dalam berinvestasi. Orang akan mencari keamanan dalam berinvestasi, sehingga mereka akan memilih investasi yang memberikan keuntungan. Dalam hal ini adalah berinvestasi dengan emas.

## 3. Naiknya permintaan emas di pasaran

Harga emas dapat naik karena permintaan akan emas dipasaran yang mengalami peningkatan.

## 4. Kurs dollar

Karena harga emas dihitung berdasarkan kurs dollar, maka jika dollar mengalami kenaikan, harga emas akan ikut terdorong naik.

## 5. Harga minyak

Pada korelasi antara emas dan minyak, mereka memiliki hubungan berbanding lurus. Jika harga minyak melonjak, hal ini menyebabkan produksi emas akan menurun dikarenakan biaya produksi semakin mahal. Penambangan emas sangatlah bergantung pada minyak untuk operasionalnya. Pada situasi lain,

ketika harga minyak melonjak naik, kinerja perusahaan akan menurun. Produksi mereka pun menurun dan bisa jadi tidak mencapai target. Hal ini menyebabkan banyak para investor dari perusahaan tersebut yang melepas kepemilikan sahamnya. Hal ini menyebabkan harga saham perusahaan itu pun melorot. Para investor pun mencari alternatif lain untuk mengembangkan financial mereka, salah satunya emas. Permintaan yang tinggi terhadap emas di saat para investor melepas kepemilikan sahamnya menyebabkan harga emas merangkak naik.

#### 6. Situasi politik dunia

Kenaikan harga emas pada akhir tahun 2002 dan awal tahun 2003 terjadi karena sekutu yang dikomando AS akan melakukan serangan ke Iraq. Pelaku pasar beralih investasi dari pasar uang dan pasar saham ke investasi emas sehingga permintaan emas melonjak tajam.

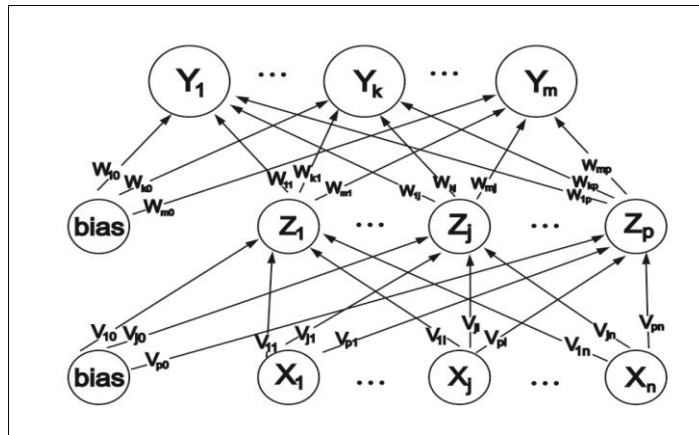


## BAB III

### PEMBAHASAN

#### A. Model *Feedforward Neural Network* dengan Algoritma *Backpropagation*

Secara umum, proses bekerjanya jaringan *neural network* menyerupai cara otak manusia memproses data input sensorik, diterima sebagai neuron input. Selanjutnya *neuron* saling berhubungan dengan sinapsis (node), dan sinyal dari neuron bekerja secara paralel digabungkan untuk menghasilkan informasi maupun reaksi (Paul, 2005 : 21). *Feedforwardneural network*(FFNN) merupakan salah satu model *neural network* yang banyak dipakai dalam berbagai bidang. Arsitektur model FFNN terdiri atas satu lapis input, satu atau lebih lapis tersembunyi, dan satu lapis output. Dalam model ini, perhitungan respon atau output  $y_k$  dilakukan dengan memproses input  $x$  mengalir dari satu lapis maju ke lapis berikutnya secara berurutan. *Single layer feedforward* dengan satu *neuron* pada lapisan tersembunyi adalah jaringan saraf yang paling dasar dan umum digunakan dalam ekonomi dan aplikasi keuangan. Kompleksitas dari arsitektur FFNN tergantung pada jumlah lapis tersembunyi dan jumlah *neuron* pada masing-masing lapis. Gambar 3.1 adalah arsitektur *feedforward neural network* dengan  $n$  buah masukan (ditambah sebuah bias), sebuah lapisan tersembunyi yang terdiri dari  $p$  unit (ditambah sebuah bias), serta  $m$  buah unit keluaran.



Sumber : (Jong, 2005 : 98)

Gambar 3.1 Arsitektur *feedforward neural network*

## 1. Algoritma Backpropagation

Algoritma pelatihan *Backpropagation* (BP) adalah salah satu algoritma dengan multilayer perceptron yang pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart dan McClelland untuk dipakai pada *Neural Network*. *Backpropagation neural network* merupakan tipe jaringan saraf tiruan yang menggunakan metode pembelajaran terawasi (Sri Kusumadewi, 2004 : 93).

Algoritma BP juga banyak dipakai pada aplikasi pengaturan karena proses pelatihannya didasarkan pada hubungan yang sederhana, yaitu jika keluaran memberikan hasil yang salah, maka penimbang dikoreksi supaya errornya dapat diperkecil dan respon jaringan selanjutnya diharapkan akan lebih mendekati harga yang benar. BP juga berkemampuan untuk memperbaiki penimbang pada lapisan tersembunyi (*hidden layer*).

Algoritma *Backpropagation* disebut sebagai propagasi balik karena ketika jaringan diberikan pola masukan sebagai pola pelatihan maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk diteruskan ke unit-unit

lapisan keluaran. Selanjutnya, unit-unit lapisan keluaran memberikan tanggapan yang disebut sebagai keluaran jaringan. Saat keluaran jaringan tidak sama dengan keluaran yang diharapkan maka keluaran akan menyebar mundur (*backward*) pada lapisan tersembunyi diteruskan ke unit pada lapisan masukan. Oleh karenanya mekanisme pelatihan tersebut dinamakan *backpropagation*.

Tahap pelatihan ini merupakan langkah bagaimana suatu jaringan saraf itu berlatih, yaitu dengan cara melakukan perubahan penimbang (sambungan antar lapisan yang membentuk jaringan melalui masing-masing unitnya). Sedangkan pemecahan masalah baru akan dilakukan jika proses pelatihan tersebut selesai, fase tersebut adalah fase *mapping* atau proses pengujian/*testing*.

Algoritma Pelatihan *Backpropagation* terdiri dari dua proses, *feedforward* dan *backpropagation* dari errornya. Algoritmanya sebagai berikut (Fausset, 1994 : 294):

Langkah 0 : Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil)

Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 – 9

Langkah 2 : untuk setiap pasang data pelatihan, lakukan langkah 3 – 8

Fase I : *Feedforward*

Langkah 3 : tiap unit masukan ( $x_i, i = 1, 2, \dots, n$ ) menerima sinyal dan meneruskannya ke unit selanjutnya yaitu lapisan tersembunyi

Langkah 4 : hitung semua keluaran pada lapisan tersembunyi ( $Z_j, j = 1, 2, \dots, p$ )

$$Z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (3.1)$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$Z_j = f(Z_{net_j}) \quad (3.2)$$

Dan kirimkan sinyal tersebut ke semua unit lapisan atasnya (unit-unit output).

Langkah ini dilakukan sebanyak jumlah lapisan tersembunyi.

Langkah 5 : hitung semua keluaran jaringan di lapisan output ( $Y_k, k = 1, 2, \dots, m$ )

$$Y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad (3.3)$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$Y_k = f(y_{net_k}) \quad (3.4)$$

Fase II : *Backpropagation*

Langkah 6 : hitung faktor  $\delta$  unit keluaran berdasarkan kesalahan di setiap unit

keluaran ( $y_k, k = 1, 2, \dots, m$ )

$$\delta_k = (t_k - y_k) f'(y_{net_k}) \quad (3.5)$$

$\delta$  merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layer di bawahnya (langkah 7)

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki  $w_{kj}$ ) dengan laju percepatan  $\alpha$

$$\Delta w_{kj} = \alpha \cdot \delta_k \cdot z_j \quad (3.6)$$

Kemudian hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{k0}$ )

$$\Delta w_{k0} = \alpha \delta_k \quad (3.7)$$

Langkah 7 : Hitung faktor  $\delta$  unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi ( $z_j, j = 1, 2, \dots, p$ )

$$\delta_{net_j} = \sum_{k=1}^m \delta_k \cdot w_{kj} \quad (3.8)$$

Faktor  $\delta$  unit tersembunyi :

$$\delta_j = \delta_{net_j} f'(z_{net_j}) \quad (3.9)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ji}$ )

$$\Delta v_{ji} = \alpha \cdot \delta_j \cdot x_i \quad (3.10)$$

Kemudian hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{j0}$ )

$$\Delta v_{j0} = \alpha \cdot \delta_j \quad (3.11)$$

Fase III : Perubahan bobot

Langkah 8 : Tiap-tiap unit output ( $Y_k, k = 1, 2, \dots, m$ ) memperbaiki bobotnya ( $j = 0, 1, 2, \dots, p$ )

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (3.12)$$

Tiap-tiap unit tersembunyi ( $Z_j, j = 1, 2, 3, \dots, p$ ) memperbaiki bobotnya ( $j = 0, 1, 2, 3, \dots, n$ )

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (3.13)$$

Langkah 9 : Kondisi pelatihan berhenti

Model FFNN algoritma *backpropagation* secara sistematis dapat dituliskan sebagai berikut :

$$y_k = \sum_{j=1}^p w_{kj} \cdot f[v_{jo} + \sum_{i=1}^n x_i v_{ji}] + w_{k0} \quad (3.14)$$

Fungsi aktivasi untuk algoritma *backpropagation* harus memiliki beberapa karakteristik penting (Fausset, 1994 : 292), yaitu :

- a. Harus kontinu
- b. Terdifferensial
- c. Monoton

Fungsi aktivasi yang memenuhi karakteristik diatas yaitu fungsi *sigmoid biner, sigmoid bipolar dan linear*.

## **2. Membangun jaringan *feedforward neural network* dengan algoritma *backpropagation***

Membangun jaringan *feedforward neural network* dengan algoritma *backpropagation* memerlukan beberapa langkah. Penulis membagi langkah-langkah tersebut kedalam beberapa tahap yaitu sebagai berikut :

- a. Menentukan input

Langkah pertama dalam menentukan jaringan *feedforward neural network* dengan algoritma *backpropagation* adalah menentukan input. Identifikasi input didasarkan pada *lag-lag* signifikan pada plot fungsi autokorelasi (ACF) dan autokorelasi parsial (PACF). Data input yang telah dipilih dinormalisasi dengan

perintah *prestd* dalam MATLAB. Fungsi aktivasi yang digunakan pada lapis tersembunyi yaitu *sigmoid biner* (tansig), sedangkan pada lapis output menggunakan fungsi aktivasi *linier* (purelin). Pembelajaran Backpropagation dilakukan dengan menentukan banyaknya neuron pada lapis tersembunyi.

#### b. Pembagian data

Data yang ada dibagi menjadi 2 yaitu *training* dan data *testing*. Beberapa komposisi data *training* dan *testing* yang sering digunakan adalah 80% untuk *training* dan 20% untuk data *testing*, 75% untuk data *training* dan 25% untuk data *testing*, atau 50% data *training* dan 50% untuk data *testing*. Komposisi ini bersifat bebas.

#### c. Normalisasi data

Sebelum melakukan pembelajaran maka data perlu dinormalisasikan. Hal ini dapat dilakukan dengan meletakkan data-data input dan target pada range tertentu. Proses normalisasi dapat dilakukan dengan bantuan mean dan standar deviasi.

##### 1) Perhitungan nilai rata-rata

$$\bar{Y} = \frac{1}{n} \sum_{t=1}^n Y_t \quad (3.15)$$

##### 2) Perhitungan nilai variansi

$$s^2 = \frac{1}{n-1} \sum_{t=1}^n (Y_t - \bar{Y})^2 \quad (3.16)$$

##### 3) Perhitungan normalisasi

$$norm = \frac{Y_t - \bar{Y}}{s} \quad (3.17)$$

dengan,

$n$  = banyaknya data

$\bar{Y}$  = rata-rata data

$Y_t$  = pengamatan pada waktu  $t$

$s^2$  = *variansi*

$s$  = simpangan baku

Proses normalisasi data dengan bantuan mean dan standar deviasi menggunakan perintah *prestd* pada MATLAB yang akan membawa data ke dalam bentuk normal. Berikut perintahnya:

```
[pn, meanp, stdp, tn, meant, stdt] = prestd (P,T)
```

dengan

P : matriks input

T : matriks target

Fungsi pada matlab akan menghasilkan:

Pn : matriks input yang ternormalisasi (mean = 0, deviasi standar = 1)

tn : matriks target yang ternormalisasi (mean = 0, deviasi standar = 1)

meanp : mean pada matriks input asli (p)

stdp : deviasi standar pada matriks input asli (p)

meant : mean pada matriks target asli (t)

stdt : deviasi standar pada matriks target asli (t)

#### d. Menentukan Model FFNN yang optimal dengan Algoritma *Backpropagation*

Sebuah jaringan harus dibentuk dengan menentukan input dari jaringan tersebut. Input diketahui dari plot ACF dan PACF yang telah dijelaskan sebelumnya. Jika input sudah diketahui, maka *neuron* pada lapis tersembunyi



harus ditentukan. Penentuan *neuron* pada lapis tersembunyi dengan cara mengestimasi. Arsitektur jaringan yang sering digunakan oleh algoritma *backpropagation* adalah jaringan *feedforward* dengan banyak lapisan. Untuk membangun suatu jaringan *feedforward* digunakan perintah *newff* pada MATLAB, yaitu

```
net = newff(PR, [S1 S2 ... SN1], {TF1 TF2 ... TFN1}, BTF, BLF, PF)
```

dengan

PR : matriks berukuran Rx2 yang berisi nilai *minimum* dan *maksimum*, dengan R adalah jumlah variabel input

S<sub>i</sub> : jumlah *neuron* pada lapisan ke-i, dengan  $i = 1, 2, \dots, N1$

TF<sub>i</sub> : fungsi aktivasi pada lapisan ke-i, dengan  $i = 1, 2, \dots, N1$

BTF : fungsi pelatihan jaringan (*default* : *trainlm*)

BLF : fungsi pelatihan untuk bobot (*default* : *learnngdm*)

PF : fungsi kinerja (default: mse)

Fungsi aktivasi TF<sub>i</sub> harus merupakan fungsi yang dapat dideferensialkan, seperti *tansig*, *logsig* atau *purelin*. Fungsi pelatihan BTF dapat digunakan fungsi-fungsi pelatihan untuk *backpropagation*, seperti *trainlm*, *trainbfg*, *trainrp* atau *traind*.

Proses membangun jaringan *feedforward neural network* dengan algoritma *backpropagation* terdiri atas :

1) Menentukan banyaknya neuron pada lapis tersembunyi

Jaringan yang dibangun akan dinilai keakuratannya dengan menentukan neuron terbaik pada lapisan tersembunyi. Indikator pemilihan penilaian yang digunakan adalah MAPE, MSE dan MAD. Berdasarkan nilai indikator yang terendah dari proses pembelajaran, maka diperoleh jaringan yang terbaik. Dalam tulisan ini, penulis menggunakan MAPE sebagai indikatornya. MAPE merupakan persentase nilai rata-rata *Absolute Error* dari kesalahan meramal tanpa menghiraukan tanda positif atau negatif yang dirumuskan (Hanke & Wichern, 2004 : 80)

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|Y_t - \hat{Y}_t|}{Y_t} \times 100\% \quad (2.19)$$

dengan,

$Y_t$  = nilai pengamatan ke-t

$\hat{Y}_t$  = nilai peramalan pada waktu ke-t

$n$  = banyaknya pengamatan

2) Menentukan input yang optimal

Jaringan yang akan dibangun seharusnya berdasarkan input yang sederhana namun optimal, untuk itu perlu dilalukan pengecekan terhadap input jaringan. Untuk mendapatkan input yang optimal perlu dilakukan pengeliminasian

terhadap input. Indikator dari optimalnya dilihat dari MAPE yang diperoleh setelah melakukan pelatihan. Input yang optimal yaitu ketika MAPE yang diperoleh sangat kecil atau paling kecil.

### 3) Menentukan bobot model

Penentuan bobot model bergantung pada pemilihan parameter pembelajaran. Pemilihan parameter pembelajaran adalah proses yang penting ketika melakukan pembelajaran. Dalam membentuk suatu jaringan, model yang kurang baik dapat diperbaiki dengan parameter-parameter secara *trial and error* untuk mendapatkan nilai bobot optimum supaya MAPE jaringan dapat diperbarui. Adapun untuk parameter-parameter yang perlu diatur ketika melakukan pembelajaran *traingdx* adalah (Sri Kusumadewi, 2004 : 156) :

#### a) Maksimum epoh

Maksimum epoh adalah jumlah epoh maksimum yang boleh dilakukan selama proses pelatihan. Iterasi akan terhenti apabila nilai epoh melebihi maksimum epoh.

Perintah di MATLAB : `net.trainParam.epochs = MaxEpoh`

Nilai default untuk maksimum epoh adalah 10

#### b) Kinerja tujuan

Kinerja tujuan adalah target nilai fungsi kinerja. Iterasi akan dihentikan apabila nilai fungsi kinerja kurang dari atau sama dengan kinerja tujuan.

Perintah di MATLAB : `net.trainParam.goal = TargetError`

Nilai default untuk kinerja tujuan adalah 0.

c) *Learning rate*

*Learning rate* adalah laju pembelajaran. Semakin besar *learning rate* akan berimplikasi pada semakin besar langkah pembelajaran.

Perintah di MATLAB : `net.trainParam.lr = LearningRate`.

Nilai default untuk *learning rate* adalah 0,01.

d) Rasio untuk menaikkan *learning rate*

Rasio yang berguna sebagai faktor pengali untuk menaikkan *learning rate* apabila *learning rate* yang ada terlalu rendah atau mencapai kekonvergenan.

Perintah di MATLAB : `net.trainParam.lr_inc = IncLearningRate`

Nilai default untuk rasio menaikkan *learning rate* adalah 1,05.

e) Rasio untuk menurunkan *learning rate*

Rasio yang berguna sebagai faktor pengali untuk menurunkan *learning rate* apabila *learning rate* yang ada terlalu tinggi atau menuju ke ketidakstabilan.

Perintah di MATLAB : `net.trainParam.lr_decc = DecLearningRate`

Nilai default untuk rasio penurunan *learning rate* adalah 0,7.

f) Maksimum kegagalan

Maksimum kegagalan diperlukan apabila pada algoritma disertai dengan validitas (optional). Maksimum kegagalan adalah ketidakvalitan terbesar yang diperbolehkan. Apabila gradient pada iterasi ke-k lebih besar daripada gradien iterasi ke-(k-1), maka kegagalannya akan bertambah 1. Iterasi akan dihentikan apabila jumlah kegagalan lebih dari maksimum kegagalan.

Perintah di MATLAB : `net.trainParam.max_fail = MaxFaile`

Nilai default untuk maksimum kegagalan adalah 5.

g) Maksimum kenaikan kerja

Maksimum kenaikan kerja adalah nilai maksimum kenaikan error yang diijinkan, antara error saat ini dan error sebelumnya.

Perintah di MATLAB : `net.trainParam.max_perf_inc = MaxPerfInc`

Nilai default untuk maksimum kenaikan kinerja adalah 1,04.

h) Gradien minimum

Gradien minimum adalah akar dari jumlah kuadrat semua gradien (bobot input, bobot lapisan, bobot bias) terkecil yang diperbolehkan. Iterasi akan dihentikan apabila nilai akar kuadrat semua gradien ini kurang dari gradien minimum.

Perintah di MATLAB : `net.trainParam.min_grad = MinGradien`

Nilai default untuk gradien minimum adalah  $10^{-10}$ .

i) Momentum

Momentum adalah perubahan bobot yang baru dengan dasar bobot sebelumnya. Besarnya momentum antara 0 sampai 1. Apabila besarnya momentum = 0 maka perubahan bobot hanya akan dipengaruhi oleh gradiennya. Sedangkan, apabila besarnya momentum = 1 maka perubahan bobot akan sama dengan perubahan bobot sebelumnya.

Perintah di MATLAB : `net.trainParam.mc = Momentum`

Nilai default untuk momentum adalah 0,9.

j) Jumlah epoh yang akan ditunjukkan kemajuannya

Parameter ini menunjukkan berapa jumlah epoh yang berselang yang akan ditunjukkan kemajuannya.

Perintah di MATLAB : `net.trainParam.show = EpochShow`

Nilai default untuk jumlah epoh yang akan ditunjukkan adalah 25.

k) Waktu maksimum untuk pelatihan

Parameter ini menunjukkan waktu maksimum yang diijinkan untuk melakukan pelatihan. Iterasi akan dihentikan apabila waktu pelatihan melebihi waktu maksimum.

Perintah di MATLAB : `net.trainParam.time = MaxTime`

Nilai default untuk waktu maksimum adalah tak terbatas (inf).

Algoritma pelatihan dilakukan untuk pengaturan bobot, sehingga pada akhir pelatihan mendapatkan bobot-bobot yang baik. Selama proses pelatihan, bobot diatur secara iteratif untuk meminimumkan fungsi kinerja jaringan. Fungsi

kinerja jaringan yang sering digunakan dalam backpropagation adalah *mean square error* (mse), fungsi ini akan mengambil rata-rata kuadrat eror yang terjadi antara output jaringan dengan target. Namun pada skripsi ini akan menggunakan nilai MAPE yang merupakan persentase nilai rata-rata *Absolute Error* dari kesalahan meramal tanpa menghiraukan tanda positif atau negatif pada sistem kerjanya.

Algoritma pelatihan yang dasar ada 2 macam (Sri Kusumadewi, 2004 : 116), yaitu:

*a. Incremental Mode*

Dalam MATLAB, perintah Incremental Mode ada 2 yaitu *learnngd* dan *learnngdm*.

*b. Batch Mode*

Dalam MATLAB, perintah *Batch Mode* ada 2 yaitu *traingd* dan *traingdm*.

Dari kedua algoritma tersebut, algoritma pelatihan dasar *Batch Mode* yang sering digunakan. Pelatihan sederhana dengan *Batch Mode* menggunakan fungsi *train* dalam matlab sebagai berikut:

```
net = train (net,P,T)
```

e) Denormalisasi

Setelah proses pelatihan selesai, maka data yang telah dinormalisasi dikembalikan seperti semula yang disebut denormalisasi data. Data akan di denormalisasi dengan fungsi *poststd* pada matlab, dengan perintah sebagai berikut :

```
[P,T]= poststd (pn, meanp, stdp, tn, meant, stdt).
```

f) Uji kesesuaian model

Untuk mengecek error pada struktur jaringan yang telah dibentuk dengan uji *white noise*. Pengujian ini dilihat dari plot ACF dan PACF dari *error training* apakah bersifat random atau tidak. Jika *error* bersifat random maka proses *white noise* terpenuhi sehingga jaringan layak digunakan untuk peramalan.

**B. Penerapan model *Feedforward Neural Network* algoritma *Backpropagation* untuk melakukan peramalan harga emas**

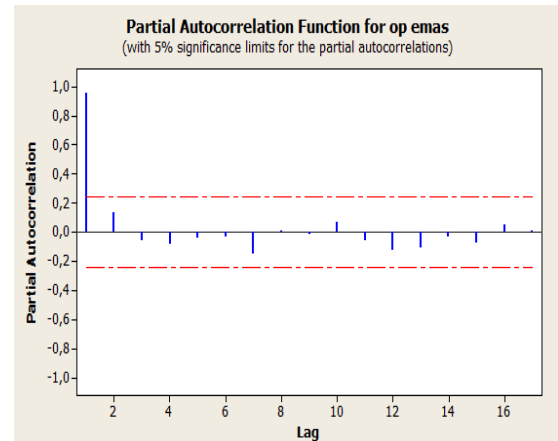
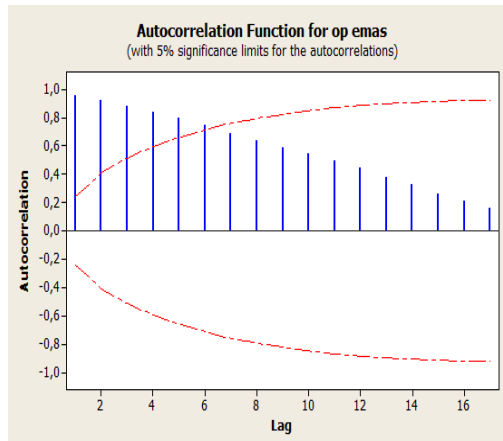
Penerapan model *Feedforward Neural Network* algoritma *Backpropagation* untuk melakukan peramalan terhadap harga open emas dengan variabel input yaitu harga open emas dan harga open minyak dunia. Data yang digunakan periode bulanan dari bulan Juli 2008 sampai bulan Februari 2014. Adapun untuk data pada lampiran I yaitu harga open emas dan harga open minyak yang digunakan diperoleh dari PT. Monex Investindo Futures cabang Yogyakarta dengan banyaknya data adalah 68 data.

**1. Membangun jaringan *feedforward neural network* dengan algoritma *backpropagation* untuk peramalan harga emas**

a. Penentuan input jaringan

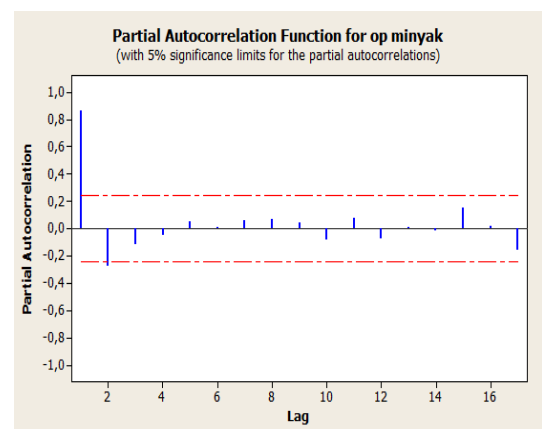
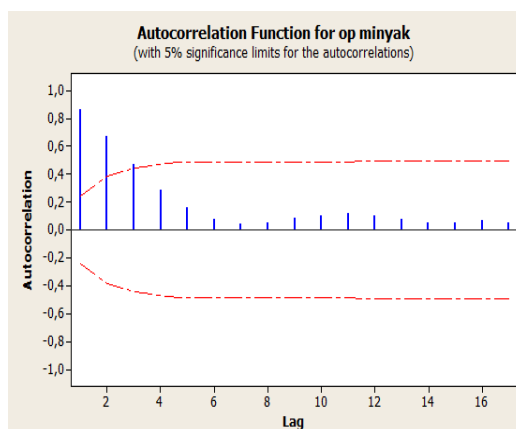
Penentuan input jaringan dilakukan dengan melihat lag-lag yang signifikan pada plot ACF dan PACF dari masing-masing variabel. Pada plot ACF dan PACF jika ada garis yang melewati selang kepercayaan (garis putus-putus merah) berarti selang tersebut telah signifikan. Plot ACF dan PACF dari masing-masing variabel diperoleh dengan bantuan MINITAB.





Gambar 3.2. Plot ACF (kiri) dan Plot PACF (kanan) data harga open emas dunia periode bulanan mulai dari bulan Juli 2008 sampai bulan Februari 2014.

Pada gambar 3.2 diatas, plot ACF untuk data harga open emas dunia yang signifikan adalah lag 1, lag 2, lag 3, lag 4, lag 5, dan lag 6. Sedangkan plot PACF untuk data harga open emas dunia yang signifikan adalah lag 1.



Gambar 3.3. Plot ACF (kiri) dan plot PACF (kanan) data harga open minyak dunia periode bulanan mulai dari bulan Juli 2008 sampai bulan Februari 2014

Untuk data harga open minyak dunia seperti yang terlihat pada gambar 3.3, plot ACF yang signifikan adalah lag 1, lag 2 dan lag 3. Sedangkan plot PACF untuk data harga open minyak dunia yang signifikan adalah lag 1 dan lag 2.

Dalam tulisan ini, penulis menggunakan variabel input berdasarkan pada plot PACF. Gambar plot PACF pada harga open emas, lag yang melewati selang kepercayaan adalah lag 1, sedangkan pada harga open minyak, lag yang melewati selang kepercayaan adalah lag 1 dan lag 2. Maka input jaringan terdiri atas  $x_1$  sebagai harga open emas lag 1,  $x_2$  sebagai harga open minyak lag 1,  $x_3$  sebagai harga open minyak lag 2 dengan banyaknya data menjadi 66 data.

#### b. Pembagian data

Data untuk peramalan menggunakan algoritma *backpropagation* dibagi menjadi 2 bagian yaitu data training dan data testing. Pada peramalan harga open emas ini, penulis menggunakan 75% data untuk data training dan 25% untuk data testing. Maka dari itu data training pada peramalan ini sebanyak 50 yang terlihat pada lampiran II dan data testingnya berjumlah 16 data yang terlihat pada lampiran III.

#### c. Normalisasi

Data Input (P) dan target (T) pada lampiran II harus dinormalisasikan terlebih dahulu. Untuk menormalisasikan data input dan target menggunakan perintah sebagai berikut :

```
[Pn, meanp, stdp, Tn, meant, stdt] = prestd (P, T)
```

Hasil proses normalisasi yang dilakukan terdapat pada lampiran IV.

d. Menentukan Model FFNN yang optimal dengan Algoritma *Backpropagation*

Untuk membangun jaringan *feedforward* yang akan digunakan untuk meramalkan harga emas menggunakan perintah *newff* pada MATLAB, yaitu :

```
net=newff(minmax(Pn),[n 1],{'tansig' 'purelin'},  
'traingdx');
```

Pada perintah tersebut, ada *n* neuron pada lapisan tersembunyi dan 1 lapisan output. Fungsi aktivasi yang digunakan pada lapisan tersembunyi adalah *tansig*, sedangkan pada lapisan output adalah *purelin*. Fungsi pelatihan yang digunakan dalam tulisan ini adalah *traingdx*.

Proses membangun jaringan *feedforward neural network* dengan algoritma *backpropagation* untuk meramalkan harga emas terdiri atas :

1) Menentukan banyaknya neuron pada lapis tersembunyi

Data input yang telah dipilih dinormalisasi dengan perintah *prestd* dalam MATLAB. Pembelajaran *Backpropagation* dilakukan dengan menentukan banyaknya *neuron* pada lapis tersembunyi dengan cara mencoba neuron lapis tersembunyi mulai dari 1 *neuron* sampai 10 *neuron* menggunakan perintah pembelajaran *traingdx*. Hasil yang diperoleh terdapat pada tabel 3.1. Model yang terbaik dipilih berdasarkan nilai MAPE yang terkecil yaitu ketika *neuron* tersembunyinya berjumlah 9 *neuron*.

Tabel 3.1. Nilai MSE dan MAPE hasil pembelajaran *traingdx* dengan algoritma *Backpropagation*

neuron tersembunyi	MSE training	MAPE training	MSE testing	MAPE testing
1	5,03E+03	4,3352	8,18E+03	13,5475
2	4,07E+03	3,7681	1,12E+04	11,7753
3	3,91E+03	4,0719	1,03E+04	12,7245
4	2,07E+03	2,964	1,98E+04	9,2626
5	2,41E+03	2,8717	1,26E+04	8,9741
6	2,59E+03	2,8867	2,20E+03	9,0211
7	1,75E+03	2,4961	4,85E+04	7,8004
8	1,15E+03	2,1217	2,77E+04	6,6303
9*	9,70E+02	1,8178	2,43E+04	5,6808
10	1,23E+03	2,0318	1,80E+04	6,3493

Keterangan: \*banyaknya neuron pada lapis tersembunyi yang terbaik

## 2) Menentukan input yang optimal

Jaringan yang akan dibangun seharusnya berdasarkan input yang sederhana namun optimal, untuk itu perlu dilalukan pengecekan terhadap input jaringan. Untuk mendapatkan input yang optimal perlu dilakukan pengeliminasian terhadap input  $x_1, x_2, x_3$ . Indikator dari optimalnya dilihat dari MAPE yang diperoleh setelah melakukan pelatihan. Input yang optimal yaitu ketika MAPE yang diperoleh sangat kecil atau paling kecil. Pada tabel (Nilai MSE dan MAPE hasil pembelajaran *traingdx* dengan algoritma *Backpropagation*), jaringan dengan 9 neuron pada lapis tersembunyi menjadi arsitektur terbaik karena menghasilkan nilai MAPE terkecil. Selanjutnya akan ditentukan input yang optimal dengan mengeliminasi input. Hasilnya sebagai berikut :

Tabel 3.2. Nilai MSE dan MAPE Input Optimal

eliminasi	MSE training	MAPE training	MSE testing	MAPE testing
.*	9,70E+02	1,8178	2,43E+04	5,6808
$X_1$	9,82E+03	5,4171	6,69E+04	16,9283
$X_2$	1,80E+03	2,7026	2,34E+04	8,4456
$X_3$	1,86E+03	2,6559	9,50E+05	8,2998

Keterangan : \*model dengan input optimal

Tabel diatas menunjukkan bahwa tanpa mengeliminasi input menghasilkan nilai MAPE pada training maupun testing terkecil, hal ini menunjukkan bahwa input  $x_1, x_2, x_3$  merupakan input yang optimal untuk meramalkan harga open emas dengan algoritma backpropagation. Dengan demikian, model backpropagation yang terbentuk adalah 9 neuron pada lapis tersembunyi dengan input  $x_1, x_2, x_3$ .

### 3) Menentukan bobot model

Parameter yang digunakan untuk meramalkan harga open emas dengan algoritma backpropagation yaitu sebagai berikut :

```
net.trainParam.epochs=5000;
```

```
net.trainParam.goal=1e-5;
```

```
net.trainParam.max_perf_inc=1.05;
```

```
net.trainParam.lr=0.2;
```

```
net.trainParam.lr_inc=1.04;
```

```
net.trainParam.lr_dec=0.6;
```

```
net.trainParam.mc=0.9;
```

```
net.trainParam.show=500;
```

Hasil bobot yang diperoleh dengan menggunakan parameter tersebut terdapat pada lampiran IX.

e. Denormalisasi

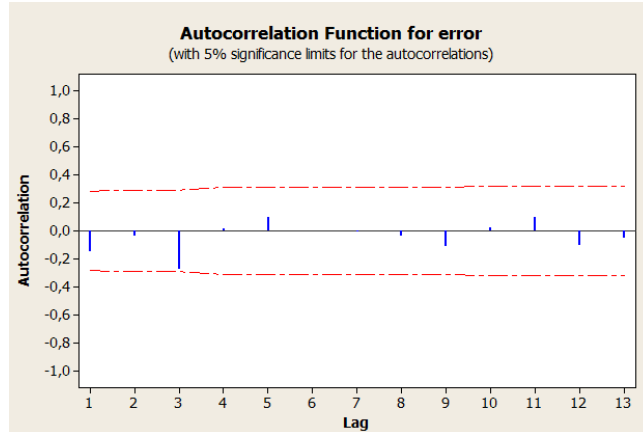
Setelah proses pelatihan selesai, maka data yang telah dinormalisasi dikembalikan seperti semula yang disebut denormalisasi data. Data akan di denormalisasi dengan fungsi *poststd* pada matlab, dengan perintah sebagai berikut:

```
[P,T]= poststd (pn, meanp, stdp, tn, meant, stdt)
```

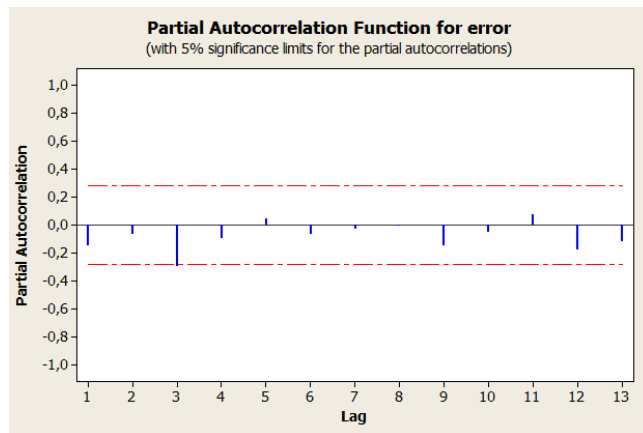
Data hasil dari proses denormalisasi kembali seperti semula pada lampiran II.

f. Uji kesesuaian model

Sebelum menggunakan model backpropagation yang terbentuk dari 9 neuron pada lapis tersembunyi dengan input  $x_1, x_2, x_3$  sebagai model peramalan harga open emas perlu dilakukan pengujian pada eror model tersebut. Pengujian ini dilakukan untuk mengetahui apakah eror pada model yang terbentuk tersebut telah white noise atau belum. Pengujian ini dilihat dari plot ACF dan PACF dari eror yang dihasilkan pada data training.



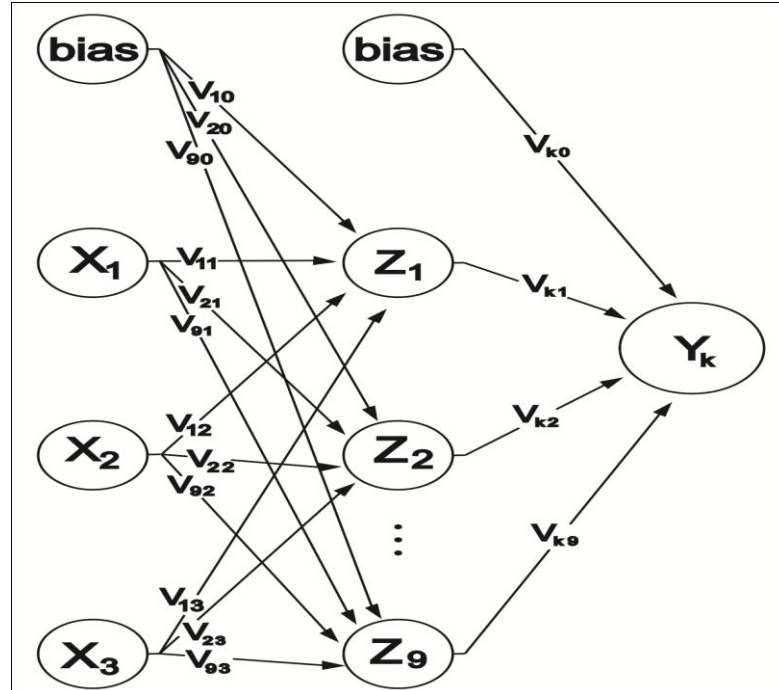
Gambar 3.4. Plot ACF model FFNN algoritma *Backpropagation* dengan 9 neuron pada lapis tersembunyi dan  $x_1, x_2, x_3$  sebagai input



Gambar 3.5. Plot PACF model FFNN algoritma *Backpropagation* dengan 9 neuron pada lapis tersembunyi dan  $x_1, x_2, x_3$  sebagai input

Dari gambar 3.4 dan gambar 3.5, ACF dan PACF terlihat bahwa semua lag berada dalam selang kepercayaan, berarti eror bersifat acak/random. Sehingga model *Feedforward Neural Network* dengan algoritma *Backpropagation* yang dibangun dari 9 neuron pada lapis tersembunyi dengan input  $x_1, x_2, x_3$  dapat digunakan sebagai model peramalan harga open emas.

Arsitektur model *Feedforward Neural Network* dengan algoritma *Backpropagation* yang dibangun dari 9 neuron pada lapis tersembunyi dengan input  $x_1, x_2, x_3$  untuk peramalan harga open emas sebagai berikut :



Gambar 3.7. Arsitektur model *Feedforward Neural Network* dengan algoritma *Backpropagation* pada peramalan harga open emas

Model *backpropagation* dengan 9 neuron tersembunyi dan input  $x_1, x_2, x_3$

secara sistematis dapat ditulis sebagai :

$$y_k = \sum_{j=1}^9 w_{kj} \cdot \frac{1 - e^{-v_{j0} + \sum_{i=1}^3 x_i v_{ji}}}{1 + e^{-v_{j0} + \sum_{i=1}^3 x_i v_{ji}}} + w_{k0}$$

## 2) Peramalan harga open emas

Proses peramalan ini menggunakan struktur terbaik yang terbangun dari 9 neuron pada lapis tersembunyi dengan input  $x_1, x_2, x_3$ . Nilai input untuk peramalan untuk bulan Maret 2014 adalah data bulan Februari 2014 yaitu US\$1242,55 per *troy ounce*. Sebelum melakukan peramalan pembelajaran data harus dinormalisasi menjadi -0,1061. Lapis output merupakan hasil prediksi algoritma *backpropagation* dengan rumus :



$$y_k = \sum_{j=1}^9 w_{kj} \cdot \frac{1 - e^{-v_{jo} + \sum_{i=1}^3 x_i v_{ji}}}{1 + e^{-v_{jo} + \sum_{i=1}^3 x_i v_{ji}}} + w_{k0}$$

Operasi keluaran lapisan input ke-j ke lapisan tersembunyi sebagai berikut :

$$\begin{aligned} z_{net_j} &= v_{jo} + \sum_{i=1}^3 x_i v_{ji} \\ &= \begin{bmatrix} -3,3514 \\ \vdots \\ 3,3925 \end{bmatrix} + \sum_{i=1}^3 (-0,1061) \begin{bmatrix} 2,0828 & 0,8575 & 0,7458 \\ \vdots & \vdots & \vdots \\ 1,0069 & 2,4043 & 1,0586 \end{bmatrix} \end{aligned}$$

Dengan menggunakan fungsi aktivasi *sigmoid bipolar*, diperoleh

$$\begin{aligned} z_j &= f(z_{net_j}) = \frac{1 - e^{-z_{net_j}}}{1 + e^{-z_{net_j}}} \\ &= f\left(\begin{bmatrix} -3,3514 \\ \vdots \\ 3,3925 \end{bmatrix} + \sum_{i=1}^3 (-0,1061) \begin{bmatrix} 2,0828 & 0,8575 & 0,7458 \\ \vdots & \vdots & \vdots \\ 1,0069 & 2,4043 & 1,0586 \end{bmatrix}\right) \\ &= \frac{1 - e^{-\begin{bmatrix} -3,3514 \\ \vdots \\ 3,3925 \end{bmatrix} + \sum_{i=1}^3 (-0,1061) \begin{bmatrix} 2,0828 & 0,8575 & 0,7458 \\ \vdots & \vdots & \vdots \\ 1,0069 & 2,4043 & 1,0586 \end{bmatrix}}}{1 + e^{-\begin{bmatrix} -3,3514 \\ \vdots \\ 3,3925 \end{bmatrix} + \sum_{i=1}^3 (-0,1061) \begin{bmatrix} 2,0828 & 0,8575 & 0,7458 \\ \vdots & \vdots & \vdots \\ 1,0069 & 2,4043 & 1,0586 \end{bmatrix}}} \\ z_j &= \begin{bmatrix} 0,9537 \\ \vdots \\ -0,8975 \end{bmatrix} \end{aligned}$$

Operasi keluaran pada lapisan tersembunyi dengan *neuron* tambahan menuju lapis

output:

$$\begin{aligned}
y_k &= y_{net_k} = w_{k0} + \sum_{j=1}^9 w_{kj} \cdot z_j \\
&= -0,6547 + \sum_{j=1}^9 [-1,5007 \quad \dots \quad 1,2975] \begin{bmatrix} 0,9537 \\ \vdots \\ -0,8975 \end{bmatrix} \\
&= -1.5762
\end{aligned}$$

Diperoleh nilai  $y_k = -1.5762$  yang kemudian di denormalisasikan menggunakan fungsi *postsd* menjadi US\$791,6 per *troy ounce*. Jadi hasil tersebut adalah hasil peramalan harga open emas pada bulan Maret menggunakan algoritma *backpropagation*. Untuk hasil peramalan pada bulan April dan Mei, dengan cara yang sama dihasilkan peramalan harga open emas pada bulan April adalah US\$1188,6 per *troy ounce* dan pada bulan Mei adalah US\$893,5 per *troy ounce*. Harga open emas yang telah diramalkan tersebut berbeda jauh dengan harga open emas yang ada. Harga open emas yang sebenarnya pada bulan Maret adalah US\$1325,85 per *troy ounce*, untuk bulan April adalah US\$1284,13 per *troy ounce*; untuk bulan Mei adalah US\$1291,14 per *troy ounce*.

Harga peramalan open emas ini adalah harga emas dunia, namun harga tersebut dapat ditransformasikan ke dalam harga emas Indonesia dengan persamaan sebagai berikut :

*harga emas Indonesia (gram)*

$$= \frac{\text{harga emas dunia}}{31,1} \times \text{kurs dollar terhadap rupiah}$$

dengan demikian peramalan harga emas di Indonesia pada bulan Maret adalah :

$$hargaemasIndonesia(gram) = \frac{791,6}{31,1} \times 11615 = 295641$$

dari hasil tersebut harga emas pada bulan Maret 2014 adalah Rp 295.641 per gram, dengan cara yang sama harga emas pada bulan April 2014 adalah Rp 430.533 per gram, sedangkan pada bulan Mei adalah Rp 331.543 per gram.

## BAB IV

### KESIMPULAN

#### A. KESIMPULAN

Berdasarkan pembahasan mengenai pembentukan model *Feedforward neural network* dengan algoritma *Backpropagation* yang diterapkan untuk meramalkan harga open emas, maka dapat disimpulkan sebagai berikut:

1. Prosedur pembentukan model *Feedforward neural network* dengan algoritma *Backpropagation* pada data *time series* terdiri atas beberapa tahap, yaitu (1) menentukan input berdasarkan plot ACF dan PACF, (2) melakukan pembagian data menjadi 2 yaitu data training dan data testing (3) menormalisasi data, (4) membangun model *Feedforward neural network* dengan algoritma *Backpropagation*, yaitu menentukan banyak neuron pada lapisan tersembunyi, menentukan input yang optimal, dan menentukan bobot model, (5) denormalisasi dan (6) uji kesesuaian model. Langkah tersebut menghasilkan model yang terbaik, yang dapat digunakan untuk peramalan.
2. Model FFNN dengan algoritma BP ini diterapkan pada data harga emas bulan Juli 2008 sampai Februari 2014 dengan variabel input yang digunakan yaitu harga emas dunia dan harga minyak dunia. Struktur jaringan terbaik yang diperoleh adalah dengan 3 neuron input dan 9 neuron pada lapis tersembunyi dengan menggunakan fungsi aktivasi *sigmoid bipolar*, fungsi *linear*, dan algoritma *traingdx*. Peramalan harga emas tersebut menghasilkan MAPE 1,8178 pada data *training* yang berarti *error* peramalan pada data *training* sebesar 1,8178 % dan 5,6808 pada data *testing* berarti *error* peramalan

pada data *testing* sebesar 5,6808%. Hasil peramalan untuk bulan Maret 2014 – Mei 2014 adalah US\$791,6 per *troy ounce*; US\$1188,6 per *troy ounce*; dan US\$893,5 per *troy ounce*.

## **B. SARAN**

Pada tulisan ini membahas tentang peramalan harga open emas menggunakan model *Feedforward neural network* dengan algoritma *Backpropagation* dengan variabel input yang digunakan yaitu harga emas dunia dan harga minyak dunia. Bagi pembaca yang tertarik menggunakan model *Feedforward neural network* dengan algoritma *Backpropagation* untuk peramalan harga open emas dapat menambahkan intervensi dari faktor-faktor penyebab perubahan harga open emas. Peramalan menggunakan model *Feedforward neural network* dengan algoritma *Backpropagation* dapat dipercepat juga dilakukan dengan berbagai macam pembelajaran lainnya misalnya *traingda* dan *trainrp*.

## DAFTAR PUSTAKA

- Abdul Halim. (2005). *Analisis Investasi*. Jakarta : Salemba Empat.
- Anonim. *Pedoman Investasi Emas*. Diakses dari [www.kebun-emas.info](http://www.kebun-emas.info) pada tanggal 12 Februari 2014.
- Crespin, Daniel. (1995). *Generalized Backpropagation*. Venezuela : Universidad Central de Venezuela.
- Desty Anna Kumalasari. (2013). *Klasifikasi metode Jaringan Syaraf Tiruan (Artificial Neural Networks) dengan fungsi Radial Basis pada Resiko Kredit*. Skripsi. Yogyakarta : FMIPA UNY.
- Dian Tri Handayani dan Agus Maman Abadi. (2012). *Penggunaan Model Neurofuzzy untuk Peramalan Nilai Tukar Rupiah terhadap Yen Jepang*. Yogyakarta : FMIPA UNY.
- Edy Suprianto. (2004). *Penerapan Jaringan Syaraf Tiruan untuk Memprediksi Harga Saham*. Skripsi. Bandung : UNIKOM.
- Fausset, L. (1994). *Fundamental of Neural Network (Archetectors, Algorithms, and Applications)*. Upper Saddle River, New-Jersey: Prentice-Hall.
- Hanke, John E and Winchern, Dean W., (2004). *Business Forecasting Eight Edition*. United States of Amerika : Pearson Education, Inc
- Heri Mauridi & Agus Kurniawan. (2006). *Supervised Neural Network dan Aplikasinya*. Yogyakarta: Graha Ilmu
- Jong Jek Siang. (2005). *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta : Penerbit ANDI.
- Liu, Derong *et al.* (2011). *Advance in Neural Network*. Berlin : Springer
- Luh PutuWidya Andriyani. (2012). *GRNN pada Peramalan data time series*. Thesis. Yogyakarta : FMIPA UGM
- Makridakis,S., Wheelwright, S.C.,& Mc Gee, V.E. (1999). *Metode dan Aplikasi Peramalan Jilid I* (Ir. Untung Sus Ardiyanto, M.Sc dan Ir. Abdul Basith, M.Sc. Terjemahan). Edisi Kedua. Jakarta: Penerbit Erlangga.

- Maya Apriyanti. (2012). *Anti Rugi dengan Berinvestasi Emas*. Yogyakarta : Pustaka Baru Press.
- McNelis, Paul D. (2005). *Neural Networks Finance*. London : Elsevier Academic Press.
- Nazari, Jamshid and Ersoy, Okan K. (1992). *Implementation of Back-propagation Neural Networks With Matlab*. Purdue University
- Raton, Boca. (2001). *Intelligent Control System Using Soft Computing Methodologies*. London : CRC Press LCC.
- Rika Wahyuni. (2005). *Neural Networks dan model ARIMA untuk Peramalan Finansial*. Thesis. Yogyakarta : FMIPA UGM.
- Rojas, Raul. (1996). *Neural Network: A Systematic Introduction*. Berlin: Springer-Verlag.
- Sielvy Evtiana. (2013). *Aplikasi Model Neuro Fuzzy untuk Memprediksi Harga Emas*. Skripsi. Yogyakarta : FMIPA UNY.
- Sri Kusumadewi. (2004). *Membangun Jaringan Syaraf Tiruan Menggunakan Matlab dan Excel Link*. Yogyakarta : Graha Ilmu
- Sudjana. (1996). *Metode Statistika*. Bandung : PT. Tarsito
- Sulistyaningsih, Ambar. (2013). *Aplikasi Model Recurrent Neural Networks dengan Variasi Kalender Islam pada Data Kunjungan Wisata Candi Prambanan*. Skripsi. Yogyakarta : FMIPA UNY.
- Tandelilin, E. (2001). *Analisis Investasi dan Manajemen Portofolio*. Yogyakarta : BPFE-YOGYAKARTA.
- Wei, W.W.S. (2006). *Time Series Analysis Univariate and Multivariate Methods Second Edition*. New York: Pearson Education.
- Widhatul Milla. (2012). *Penerapan model Neural Networks dengan Algoritma Recurren Sebagai Metode Peramalan Harga Koin Emas di Pegadaian*. Skripsi. Yogyakarta : FMIPA UNY.

## LAMPIRAN I

Data harga open emas dunia dan harga open minyak dunia

(Juli 2018 – Februari 2014)

No	tanggal	open emas US\$ per troy ounce	open minyak US\$ per barrel
1	01-Jul-08	926,2	129,92
2	01-Agust-08	913,2	124,06
3	01-Sep-08	833,2	116,71
4	01-Okt-08	871,6	101,84
5	01-Nop-08	726,7	67,39
6	01-Des-08	818,4	53,89
7	01-Jan-09	884,4	43,49
8	01-Feb-09	924,6	41,67
9	01-Mar-09	939,1	44,24
10	01-Apr-09	918	48,65
11	01-Mei-09	886,5	50,71
12	01-Jun-09	978,3	66,37
13	01-Jul-09	926	70,51
14	01-Agust-09	952,8	69,29
15	01-Sep-09	950,4	69,57
16	01-Okt-09	1007,1	70,19
17	01-Nop-09	1042,6	76,99
18	01-Des-09	1179,3	77,35
19	01-Jan-10	1097,4	79,61
20	01-Feb-10	1081,3	72,69
21	01-Mar-10	1118,2	79,78
22	01-Apr-10	1113,4	83,34
23	01-Mei-10	1179,8	86,2
24	01-Jun-10	1215,9	74,46
25	01-Jul-10	1241,8	75,19
26	01-Agust-10	1181,25	79,03
27	01-Sep-10	1247,95	71,72
28	01-Okt-10	1307,3	79,78
29	01-Nop-10	1359,95	81,37
30	01-Des-10	1385,85	83,67
31	01-Jan-11	1418,5	91,27



No	tanggal	open emas US\$ per troy ounce	open minyak US\$ per barrel
32	01-Feb-11	1332,3	92,04
33	01-Mar-11	1410,85	96,99
34	01-Apr-11	1432,35	106,64
35	01-Mei-11	1571,8	113,82
36	01-Jun-11	1535,35	102,61
37	01-Jul-11	1500,7	95,04
38	01-Agust-11	1607,05	96,4
39	01-Sep-11	1825,1	88,73
40	01-Okt-11	1619,2	78,46
41	01-Nop-11	1714,5	92,58
42	01-Des-11	1747,9	100,4
43	01-Jan-12	1568,3	100,25
44	01-Feb-12	1736,4	98,28
45	01-Mar-12	1694,7	106,86
46	01-Apr-12	1673,95	103,3
47	01-Mei-12	1664,65	104,88
48	01-Jun-12	1559,95	86,43
49	01-Jul-12	1598,25	84,74
50	01-Agust-12	1613,6	87,98
51	01-Sep-12	1690,9	96,36
52	01-Okt-12	1770,35	92,07
53	01-Nop-12	1719,7	86,07
54	01-Des-12	1713,85	88,93
55	01-Jan-13	1673,5	91,73
56	01-Feb-13	1664	97,41
57	01-Mar-13	1579,15	91,72
58	01-Apr-13	1597,31	97,22
59	01-Mei-13	1475,14	93,04
60	01-Jun-13	1389,22	91,73
61	01-Jul-13	1234,48	96,5
62	01-Agust-13	1323,48	105,24
63	01-Sep-13	1393,09	106,95
64	01-Okt-13	1327,35	102,28
65	01-Nop-13	1322,89	96,27
66	01-Des-13	1250,03	92,67
67	01-Jan-14	1209,52	98,63
68	01-Feb-14	1242,55	97,32

## LAMPIRAN II

### Data Training

No	Target	Input		
	T	x1	x2	x3
1	833,2	913,2	124,06	129,92
2	871,6	833,2	116,71	124,06
3	726,7	871,6	101,84	116,71
4	818,4	726,7	67,39	101,84
5	884,4	818,4	53,89	67,39
6	924,6	884,4	43,49	53,89
7	939,1	924,6	41,67	43,49
8	918	939,1	44,24	41,67
9	886,5	918	48,65	44,24
10	978,3	886,5	50,71	48,65
11	926	978,3	66,37	50,71
12	952,8	926	70,51	66,37
13	950,4	952,8	69,29	70,51
14	1007,1	950,4	69,57	69,29
15	1042,6	1007,1	70,19	69,57
16	1179,3	1042,6	76,99	70,19
17	1097,4	1179,3	77,35	76,99
18	1081,3	1097,4	79,61	77,35
19	1118,2	1081,3	72,69	79,61
20	1113,4	1118,2	79,78	72,69
21	1179,8	1113,4	83,34	79,78
22	1215,9	1179,8	86,2	83,34
23	1241,8	1215,9	74,46	86,2
24	1181,25	1241,8	75,19	74,46
25	1247,95	1181,25	79,03	75,19
26	1307,3	1247,95	71,72	79,03
27	1359,95	1307,3	79,78	71,72
28	1385,85	1359,95	81,37	79,78
29	1418,5	1385,85	83,67	81,37
30	1332,3	1418,5	91,27	83,67
31	1410,85	1332,3	92,04	91,27
32	1432,35	1410,85	96,99	92,04

No	Target	Input		
	T	x1	x2	x3
33	1571,8	1432,35	106,64	96,99
34	1535,35	1571,8	113,82	106,64
35	1500,7	1535,35	102,61	113,82
36	1607,05	1500,7	95,04	102,61
37	1825,1	1607,05	96,4	95,04
38	1619,2	1825,1	88,73	96,4
39	1714,5	1619,2	78,46	88,73
40	1747,9	1714,5	92,58	78,46
41	1568,3	1747,9	100,4	92,58
42	1736,4	1568,3	100,25	100,4
43	1694,7	1736,4	98,28	100,25
44	1673,95	1694,7	106,86	98,28
45	1664,65	1673,95	103,3	106,86
46	1559,95	1664,65	104,88	103,3
47	1598,25	1559,95	86,43	104,88
48	1613,6	1598,25	84,74	86,43
49	1690,9	1613,6	87,98	84,74
50	1770,35	1690,9	96,36	87,98

### LAMPIRAN III

#### *Data Testing*

No	Target	Input		
	T	x1	x2	x3
1	1719,7	1770,35	92,07	96,36
2	1713,85	1719,7	86,07	92,07
3	1673,5	1713,85	88,93	86,07
4	1664	1673,5	91,73	88,93
5	1579,15	1664	97,41	91,73
6	1597,31	1579,15	91,72	97,41
7	1475,14	1597,31	97,22	91,72
8	1389,22	1475,14	93,04	97,22
9	1234,48	1389,22	91,73	93,04
10	1323,48	1234,48	96,5	91,73
11	1393,09	1323,48	105,24	96,5
12	1327,35	1393,09	106,95	105,24
13	1322,89	1327,35	102,28	106,95
14	1250,03	1322,89	96,27	102,28
15	1209,52	1250,03	92,67	96,27
16	1242,55	1209,52	98,63	92,67

#### LAMPIRAN IV

Hasil normalisasi data

NO	Tn	Pn		
		x1	x2	x3
1	-1,4454	-1,1515	2,13135	2,27989
2	-1,3247	-1,4054	1,74724	1,98928
3	-1,78	-1,2836	0,97014	1,62477
4	-1,4919	-1,7435	-0,8302	0,88733
5	-1,2844	-1,4524	-1,5357	-0,8211
6	-1,1581	-1,2429	-2,0792	-1,4906
7	-1,1125	-1,1153	-2,1744	-2,0064
8	-1,1789	-1,0693	-2,04	-2,0967
9	-1,2778	-1,1363	-1,8096	-1,9692
10	-0,9894	-1,2363	-1,7019	-1,7505
11	-1,1537	-0,9449	-0,8835	-1,6483
12	-1,0695	-1,1109	-0,6672	-0,8717
13	-1,077	-1,0258	-0,7309	-0,6664
14	-0,8988	-1,0334	-0,7163	-0,7269
15	-0,7873	-0,8535	-0,6839	-0,713
16	-0,3577	-0,7408	-0,3285	-0,6823
17	-0,6151	-0,3069	-0,3097	-0,345
18	-0,6657	-0,5668	-0,1916	-0,3272
19	-0,5497	-0,6179	-0,5532	-0,2151
20	-0,5648	-0,5008	-0,1827	-0,5583
21	-0,3561	-0,516	0,00332	-0,2067
22	-0,2427	-0,3053	0,15279	-0,0301
23	-0,1613	-0,1907	-0,4607	0,1117
24	-0,3516	-0,1085	-0,4226	-0,4705
25	-0,1419	-0,3007	-0,2219	-0,4343
26	0,04458	-0,0889	-0,6039	-0,2439
27	0,21004	0,09944	-0,1827	-0,6064
28	0,29143	0,26656	-0,0996	-0,2067
29	0,39404	0,34877	0,02057	-0,1278
30	0,12314	0,45241	0,41775	-0,0138
31	0,37	0,1788	0,45799	0,36314
32	0,43757	0,42813	0,71667	0,40132

NO	Tn	Pn		
		x1	x2	x3
33	0,87581	0,49637	1,22098	0,64681
34	0,76126	0,93902	1,59621	1,12538
35	0,65237	0,82332	1,01038	1,48145
36	0,98659	0,71333	0,61477	0,92552
37	1,67184	1,05091	0,68584	0,5501
38	1,02477	1,74304	0,28501	0,61755
39	1,32427	1,08947	-0,2517	0,23717
40	1,42923	1,39197	0,48621	-0,2721
41	0,86481	1,49799	0,89488	0,4281
42	1,39309	0,92791	0,88704	0,81592
43	1,26204	1,46149	0,78409	0,80848
44	1,19683	1,32912	1,23248	0,71078
45	1,1676	1,26326	1,04643	1,13629
46	0,83857	1,23374	1,12901	0,95974
47	0,95893	0,9014	0,16481	1,03809
48	1,00717	1,02297	0,07649	0,12311
49	1,2501	1,0717	0,24581	0,0393
50	1,49978	1,31706	0,68375	0,19998

## LAMPIRAN V

Program Feedforward neural network dengan algoritma Backpropagation

menggunakan MATLAB ( dengan fungsi aktivasi sigmoid bipolar pada hidden

layer, fungsi linear pada lapisan output dan menggunakan pembelajaran traindx)

```
>> P=[913.2      833.2 871.6 726.7 818.4 884.4 924.6 939.1 918
      886.5 978.3 926   952.8 950.4 1007.1      1042.6
      1179.3      1097.4      1081.3      1118.2      1113.4
      1179.8      1215.9      1241.8      1181.25      1247.95
      1307.3      1359.95      1385.85      1418.5      1332.3
      1410.85      1432.35      1571.8      1535.35      1500.7
      1607.05      1825.1      1619.2      1714.5      1747.9
      1568.3      1736.4      1694.7      1673.95      1664.65
      1559.95      1598.25      1613.6      1690.9

124.06      116.71      101.84      67.39 53.89 43.49 41.67 44.24
48.65 50.71 66.37 70.51 69.29 69.57 70.19 76.99 77.35 79.61
72.69 79.78 83.34 86.2   74.46 75.19 79.03 71.72 79.78 81.37
83.67 91.27 92.04 96.99 106.64      113.82      102.61
95.04 96.4   88.73 78.46 92.58 100.4 100.25      98.28
106.86      103.3 104.88      86.43 84.74 87.98 96.36

129.92      124.06      116.71      101.84      67.39 53.89 43.49
41.67 44.24 48.65 50.71 66.37 70.51 69.29 69.57 70.19 76.99
77.35 79.61 72.69 79.78 83.34 86.2   74.46 75.19 79.03 71.72
79.78 81.37 83.67 91.27 92.04 96.99 106.64      113.82
102.61      95.04 96.4   88.73 78.46 92.58 100.4 100.25
98.28 106.86      103.3 104.88      86.43 84.74 87.98

];

T=[833.2      871.6 726.7 818.4 884.4 924.6 939.1 918   886.5 978.3
   926   952.8 950.4 1007.1      1042.6      1179.3
   1097.4      1081.3      1118.2      1113.4      1179.8
   1215.9      1241.8      1181.25      1247.95      1307.3
   1359.95      1385.85      1418.5      1332.3      1410.85
   1432.35      1571.8      1535.35      1500.7      1607.05
   1825.1      1619.2      1714.5      1747.9      1568.3
   1736.4      1694.7      1673.95      1664.65      1559.95
   1598.25      1613.6      1690.9      1770.35

];

[m, n]=size(P);

[Pn,meanp, stdp,Tn,meant, stdt]=prestd(P,T);

net=newff(minmax(Pn),[9 1],{ 'tansig' 'purelin' },
'traingdx');
```

```

BobotAwal_Input=net.IW{1,1}
BobotAwal_Bias_Input=net.b{1,1}
BobotAwal_Lapisan=net.LW{2,1}
BobotAwal_Bias_Lapisan=net.b{2,1}
net.trainParam.epochs=5000;
net.trainParam.goal=1e-5;
net.trainParam.max_perf_inc=1.05;
net.trainParam.lr=0.2;
net.trainParam.lr_inc=1.04;
net.trainParam.lr_dec=0.6;net.trainParam.mc=0.9;
net.trainParam.show=500;
net=train(net,Pn,Tn);
BobotAkhir_Input=net.IW{1,1}
BobotAkhir_Bias_Input=net.b{1,1}
BobotAkhir_Lapisan=net.LW{2,1}
BobotAkhir_Bias_Lapisan=net.b{2,1}
ab=sim(net,Pn);
a=poststd(ab,meant,stdt);
E=T-a;
MSE=mse(E)
mape=[abs((T-a)./T).*100];
MAPE=sum(mape)/50
Q=[1770.35 1719.7      1713.85      1673.5      1664  1579.15
    1597.31      1475.14      1389.22      1234.48      1323.48
    1393.09      1327.35      1322.89      1250.03      1209.52

92.07 86.07 88.93 91.73 97.41 91.72 97.22 93.04 91.73 96.5
    105.24      106.95      102.28      96.27 92.67 98.63

96.36 92.07 86.07 88.93 91.73 97.41 91.72 97.22 93.04 91.73 96.5
    105.24      106.95      102.28      96.27 92.67

];

```



```

TQ=[1719.7 1713.85    1673.5    1664 1579.15    1597.31
     1475.14    1389.22    1234.48    1323.48    1393.09
     1327.35    1322.89    1250.03    1209.52    1242.55

];

Qn=trastd(Q,meanp,stdp);

bn=sim(net,Qn);

b=poststd(bn,meant,stdt);

E1=TQ-b

MSE1=mse(E1)

mape1=[abs((TQ-b)./TQ).*100];

MAPE1=sum(mape)/16

```

## LAMPIRAN VI

Program *Feedforward neural network* dengan algoritma *Backpropagation* untuk mencari input optimum dengan mengeliminasi  $x_1$  menggunakan MATLAB (dengan fungsi aktivasi sigmoid bipolar pada hidden layer dengan 9 neuron, fungsi linear pada lapisan output dan menggunakan pembelajaran *traingdx*)

---

```
>> P=[124.06      116.71      101.84      67.39 53.89 43.49 41.67
      44.24 48.65 50.71 66.37 70.51 69.29 69.57 70.19 76.99 77.35
      79.61 72.69 79.78 83.34 86.2   74.46 75.19 79.03 71.72 79.78
      81.37 83.67 91.27 92.04 96.99 106.64      113.82
      102.61      95.04 96.4   88.73 78.46 92.58 100.4 100.25
      98.28 106.86      103.3 104.88      86.43 84.74 87.98 96.36

      129.92      124.06      116.71      101.84      67.39 53.89 43.49
      41.67 44.24 48.65 50.71 66.37 70.51 69.29 69.57 70.19 76.99
      77.35 79.61 72.69 79.78 83.34 86.2   74.46 75.19 79.03 71.72
      79.78 81.37 83.67 91.27 92.04 96.99 106.64      113.82
      102.61      95.04 96.4   88.73 78.46 92.58 100.4 100.25
      98.28 106.86      103.3 104.88      86.43 84.74 87.98

];

>> T=[833.2      871.6 726.7 818.4 884.4 924.6 939.1 918   886.5
      978.3 926   952.8 950.4 1007.1      1042.6      1179.3
      1097.4      1081.3      1118.2      1113.4      1179.8
      1215.9      1241.8      1181.25      1247.95      1307.3
      1359.95      1385.85      1418.5      1332.3      1410.85
      1432.35      1571.8      1535.35      1500.7      1607.05
      1825.1      1619.2      1714.5      1747.9      1568.3
      1736.4      1694.7      1673.95      1664.65      1559.95
      1598.25      1613.6      1690.9      1770.35

];

[m, n]=size(P);

[Pn,meanp, stdp,Tn,meant, stdt]=prestd(P,T);

net=newff(minmax(Pn),[9 1],{ 'tansig' 'purelin' },
'traingdx');

BobotAwal_Input=net.IW{1,1}

BobotAwal_Bias_Input=net.b{1,1}

BobotAwal_Lapisan=net.LW{2,1}
```

```

BobotAwal_Bias_Lapisan=net.b{2,1}
net.trainParam.epochs=5000;
net.trainParam.goal=1e-5;
net.trainParam.max_perf_inc=1.05;
net.trainParam.lr=0.2;
net.trainParam.lr_inc=1.04;
net.trainParam.lr_dec=0.6;net.trainParam.mc=0.9;
net.trainParam.show=500;
net=train(net,Pn,Tn);
BobotAkhir_Input=net.IW{1,1}
BobotAkhir_Bias_Input=net.b{1,1}
BobotAkhir_Lapisan=net.LW{2,1}
BobotAkhir_Bias_Lapisan=net.b{2,1}
ab=sim(net,Pn);
a=poststd(ab,meant,stdt);
E=T-a;
MSE=mse(E)
mape=[abs((T-a)./T).*100];
MAPE=sum(mape)/50
>> Q=[92.07      86.07 88.93 91.73 97.41 91.72 97.22 93.04 91.73
      96.5  105.24      106.95      102.28      96.27 92.67 98.63
      96.36 92.07 86.07 88.93 91.73 97.41 91.72 97.22 93.04 91.73 96.5
      105.24      106.95      102.28      96.27 92.67
      ];
>> TQ=[1719.7      1713.85      1673.5      1664  1579.15
      1597.31      1475.14      1389.22      1234.48      1323.48
      1393.09      1327.35      1322.89      1250.03      1209.52
      1242.55
      ];
Qn=trastd(Q,meanp,stdp);
bn=sim(net,Qn);

```

```
b=poststd(bn,meant,stdt);  
E1=TQ-b  
MSE1=mse(E1)  
mape1=[abs((TQ-b)./TQ).*100];  
MAPE1=sum(mape)/16
```

## LAMPIRAN VII

Program *Feedforward neural network* dengan algoritma *Backpropagation* untuk mencari input optimum dengan mengeliminasi  $x_2$  menggunakan MATLAB (dengan fungsi aktivasi sigmoid bipolar pada hidden layer dengan 9 neuron, fungsi linear pada lapisan output dan menggunakan pembelajaran *traingdx*)

---

```
>> P=[913.2      833.2 871.6 726.7 818.4 884.4 924.6 939.1 918
      886.5 978.3 926   952.8 950.4 1007.1      1042.6
      1179.3      1097.4      1081.3      1118.2      1113.4
      1179.8      1215.9      1241.8      1181.25      1247.95
      1307.3      1359.95      1385.85      1418.5      1332.3
      1410.85      1432.35      1571.8      1535.35      1500.7
      1607.05      1825.1      1619.2      1714.5      1747.9
      1568.3      1736.4      1694.7      1673.95      1664.65
      1559.95      1598.25      1613.6      1690.9

; 129.92   124.06      116.71      101.84      67.39 53.89 43.49
   41.67 44.24 48.65 50.71 66.37 70.51 69.29 69.57 70.19 76.99
   77.35 79.61 72.69 79.78 83.34 86.2   74.46 75.19 79.03 71.72
   79.78 81.37 83.67 91.27 92.04 96.99 106.64      113.82
   102.61      95.04 96.4   88.73 78.46 92.58 100.4 100.25
   98.28 106.86      103.3 104.88      86.43 84.74 87.98];

>> T=[833.2      871.6 726.7 818.4 884.4 924.6 939.1 918   886.5
      978.3 926   952.8 950.4 1007.1      1042.6      1179.3
      1097.4      1081.3      1118.2      1113.4      1179.8
      1215.9      1241.8      1181.25      1247.95      1307.3
      1359.95      1385.85      1418.5      1332.3      1410.85
      1432.35      1571.8      1535.35      1500.7      1607.05
      1825.1      1619.2      1714.5      1747.9      1568.3
      1736.4      1694.7      1673.95      1664.65      1559.95
      1598.25      1613.6      1690.9      1770.35

];

[m, n]=size(P);

[Pn,meanp,stdp,Tn,meant,stdt]=prestd(P,T);

net=newff(minmax(Pn),[9 1],{ 'tansig' 'purelin' },
'traingdx');

BobotAwal_Input=net.IW{1,1}

BobotAwal_Bias_Input=net.b{1,1}

BobotAwal_Lapisan=net.LW{2,1}
```

```

BobotAwal_Bias_Lapisan=net.b{2,1}
net.trainParam.epochs=5000;
net.trainParam.goal=1e-5;
net.trainParam.max_perf_inc=1.05;
net.trainParam.lr=0.2;
net.trainParam.lr_inc=1.04;
net.trainParam.lr_dec=0.6;net.trainParam.mc=0.9;
net.trainParam.show=500;
net=train(net,Pn,Tn);
BobotAkhir_Input=net.IW{1,1}
BobotAkhir_Bias_Input=net.b{1,1}
BobotAkhir_Lapisan=net.LW{2,1}
BobotAkhir_Bias_Lapisan=net.b{2,1}
ab=sim(net,Pn);
a=poststd(ab,meant,stdt);
E=T-a;
MSE=mse(E)
mape=[abs((T-a)./T).*100];
MAPE=sum(mape)/50
>> Q=[1770.35    1719.7    1713.85    1673.5    1664
      1579.15    1597.31    1475.14    1389.22    1234.48
      1323.48    1393.09    1327.35    1322.89    1250.03
      1209.52

; 96.36    92.07 86.07 88.93 91.73 97.41 91.72 97.22 93.04 91.73
   96.5 105.24    106.95    102.28    96.27 92.67

];

>> TQ=[1719.7    1713.85    1673.5    1664 1579.15
      1597.31    1475.14    1389.22    1234.48    1323.48
      1393.09    1327.35    1322.89    1250.03    1209.52
      1242.55

];

```

```

Qn=trastd(Q,meanp,stdp);
bn=sim(net,Qn);
b=poststd(bn,meant,stdt);
E1=TQ-b
MSE1=mse(E1)
mape1=[abs((TQ-b)./TQ).*100];
MAPE1=sum(mape)/16

```

## LAMPIRAN VIII

Program *Feedforward neural network* dengan algoritma *Backpropagation* untuk mencari input optimum dengan mengeliminasi  $x_3$  menggunakan MATLAB (dengan fungsi aktivasi sigmoid bipolar pada hidden layer dengan 9 neuron, fungsi linear pada lapisan output dan menggunakan pembelajaran *traingdx*)

---

```
>> P=[913.2      833.2 871.6 726.7 818.4 884.4 924.6 939.1 918
      886.5 978.3 926   952.8 950.4 1007.1      1042.6
      1179.3      1097.4      1081.3      1118.2      1113.4
      1179.8      1215.9      1241.8      1181.25     1247.95
      1307.3      1359.95     1385.85     1418.5      1332.3
      1410.85     1432.35     1571.8      1535.35     1500.7
      1607.05     1825.1      1619.2      1714.5      1747.9
      1568.3      1736.4      1694.7      1673.95     1664.65
      1559.95     1598.25     1613.6      1690.9

      124.06      116.71      101.84      67.39 53.89 43.49 41.67 44.24
      48.65 50.71 66.37 70.51 69.29 69.57 70.19 76.99 77.35 79.61
      72.69 79.78 83.34 86.2  74.46 75.19 79.03 71.72 79.78 81.37
      83.67 91.27 92.04 96.99 106.64      113.82      102.61
      95.04 96.4  88.73 78.46 92.58 100.4 100.25      98.28
      106.86      103.3 104.88      86.43 84.74 87.98 96.36

];

>> T=[833.2      871.6 726.7 818.4 884.4 924.6 939.1 918   886.5
      978.3 926   952.8 950.4 1007.1      1042.6      1179.3
      1097.4      1081.3      1118.2      1113.4      1179.8
      1215.9      1241.8      1181.25     1247.95     1307.3
      1359.95     1385.85     1418.5      1332.3      1410.85
      1432.35     1571.8      1535.35     1500.7      1607.05
      1825.1      1619.2      1714.5      1747.9      1568.3
      1736.4      1694.7      1673.95     1664.65     1559.95
      1598.25     1613.6      1690.9      1770.35

];

[m, n]=size(P);

[Pn,meanp,stdp,Tn,meant,stdt]=prestd(P,T);

net=newff(minmax(Pn),[9 1],{ 'tansig' 'purelin' },
'traingdx');

BobotAwal_Input=net.IW{1,1}

BobotAwal_Bias_Input=net.b{1,1}
```



```

BobotAwal_Lapisan=net.LW{2,1}
BobotAwal_Bias_Lapisan=net.b{2,1}
net.trainParam.epochs=5000;
net.trainParam.goal=1e-5;
net.trainParam.max_perf_inc=1.05;
net.trainParam.lr=0.2;
net.trainParam.lr_inc=1.04;
net.trainParam.lr_dec=0.6;net.trainParam.mc=0.9;
net.trainParam.show=500;
net=train(net,Pn,Tn);
BobotAkhir_Input=net.IW{1,1}
BobotAkhir_Bias_Input=net.b{1,1}
BobotAkhir_Lapisan=net.LW{2,1}
BobotAkhir_Bias_Lapisan=net.b{2,1}
ab=sim(net,Pn);
a=poststd(ab,meant,stdt);
E=T-a;
MSE=mse(E)
mape=[abs((T-a)./T).*100];
MAPE=sum(mape)/50
>> Q=[92.07      86.07 88.93 91.73 97.41 91.72 97.22 93.04 91.73
      96.5  105.24      106.95      102.28      96.27 92.67 98.63
      96.36 92.07 86.07 88.93 91.73 97.41 91.72 97.22 93.04 91.73 96.5
      105.24      106.95      102.28      96.27 92.67
      ];
>> TQ=[1719.7      1713.85      1673.5      1664  1579.15
      1597.31      1475.14      1389.22      1234.48      1323.48
      1393.09      1327.35      1322.89      1250.03      1209.52
      1242.55
      ];
Qn=trastd(Q,meanp,stdp);

```

```
bn=sim(net,Qn);  
b=poststd(bn,meant,stdt);  
E1=TQ-b  
MSE1=mse(E1)  
mape1=[abs((TQ-b)./TQ).*100];  
MAPE1=sum(mape)/16
```

## LAMPIRAN IX

### Hasil pembobotan

BobotAwal\_Input =

0.8307	0.9936	0.6144
0.9654	-0.6596	0.8709
-1.0044	1.0261	0.3341
0.8949	0.8014	-0.8007
0.5917	-0.0529	1.2434
-1.0130	0.6120	0.8702
-0.8089	-1.0590	0.5200
0.2864	-0.3871	1.2545
1.1505	0.8466	0.4871

BobotAwal\_Bias\_Input =

-2.9469
-2.2778
1.4473
-0.6372
-0.1149
-0.7948
-1.5266
2.0609
2.8860

BobotAwal\_Lapisan =

-0.2155	0.3110	-0.6576	0.4121	-0.9363	-0.4462
-0.9077	-0.8057	0.6469			

BobotAwal\_Bias\_Lapisan =

0.3897
--------

BobotAkhir\_Input =

2.0828	0.8575	0.7458
1.1012	-0.8556	0.6249
0.0425	1.1504	0.5714
2.5867	1.3212	0.2116
0.4173	-0.5446	2.0700
-0.9608	0.7332	0.7651
-1.6132	-3.0038	-0.1824
0.9430	-0.7218	0.4033
1.0069	2.4043	1.0586

BobotAkhir\_Bias\_Input =

-3.3514
---------

```
-2.2570
 0.9141
-3.2954
-0.4768
-1.9356
-0.8703
 1.9947
 3.3925
BobotAkhir_Lapisan =
  -1.5007   -0.3422   -1.6651    1.5327    0.7299   -1.7882
-0.7666   -1.5707    1.2975
BobotAkhir_Bias_Lapisan =
  -0.6547
```

## LAMPIRAN X

Program *Feedforward neural network* dengan algoritma *Backpropagation* untuk peramalan harga open emas menggunakan MATLAB ( dengan fungsi aktivasi sigmoid bipolar pada hidden layer, fungsi linear pada lapisan output dan menggunakan pembelajaran *traingdx* dengan 9 neuron tersembunyi dan  $x_1, x_2, x_3$

sebagai input)

---

```
>> P=[913.2      833.2      871.6      726.7      818.4
      884.4      924.6      939.1      918      886.5      978.3
      926      952.8      950.4      1007.1      1042.6
      1179.3      1097.4      1081.3      1118.2      1113.4
      1179.8      1215.9      1241.8      1181.25      1247.95
      1307.3      1359.95      1385.85      1418.5      1332.3
      1410.85      1432.35      1571.8      1535.35      1500.7
      1607.05      1825.1      1619.2      1714.5      1747.9
      1568.3      1736.4      1694.7      1673.95      1664.65
      1559.95      1598.25      1613.6      1690.9

124.06      116.71      101.84      67.39      53.89      43.49
      41.67      44.24      48.65      50.71      66.37
      70.51      69.29      69.57      70.19      76.99
      77.35      79.61      72.69      79.78      83.34
      86.2      74.46      75.19      79.03      71.72      79.78
      81.37      83.67      91.27      92.04      96.99
      106.64      113.82      102.61      95.04      96.4      88.73
      78.46      92.58      100.4      100.25      98.28
      106.86      103.3      104.88      86.43      84.74
      87.98      96.36

129.92      124.06      116.71      101.84      67.39      53.89
      43.49      41.67      44.24      48.65      50.71
      66.37      70.51      69.29      69.57      70.19
      76.99      77.35      79.61      72.69      79.78
      83.34      86.2      74.46      75.19      79.03      71.72
      79.78      81.37      83.67      91.27      92.04
      96.99      106.64      113.82      102.61      95.04
      96.4      88.73      78.46      92.58      100.4
      100.25      98.28      106.86      103.3      104.88
      86.43      84.74      87.98

];
```

```

T=[833.2  871.6      726.7      818.4      884.4      924.6
    939.1    918  886.5      978.3      926  952.8
    950.4    1007.1    1042.6    1179.3    1097.4
    1081.3    1118.2    1113.4    1179.8    1215.9
    1241.8    1181.25    1247.95    1307.3    1359.95
    1385.85    1418.5    1332.3    1410.85    1432.35
    1571.8    1535.35    1500.7    1607.05    1825.1
    1619.2    1714.5    1747.9    1568.3    1736.4
    1694.7    1673.95    1664.65    1559.95    1598.25
    1613.6    1690.9    1770.35

```

```

];

[m, n]=size(P);

[Pn,meanp,stdp,Tn,meant,stdt]=prestd(P,T);

net=newff(minmax(Pn),[9 1],{ 'tansig' 'purelin' },
'traingdx');

BobotAwal_Input=net.IW{1,1}
BobotAwal_Bias_Input=net.b{1,1}
BobotAwal_Lapisan=net.LW{2,1}
BobotAwal_Bias_Lapisan=net.b{2,1}

net.trainParam.epochs=5000;
net.trainParam.goal=1e-5;
net.trainParam.max_perf_inc=1.05;
net.trainParam.lr=0.2;
net.trainParam.lr_inc=1.04;
net.trainParam.lr_dec=0.6;net.trainParam.mc=0.9;
net.trainParam.show=500;

net=train(net,Pn,Tn);

vji=net.IW{1,1}
vj0=net.b{1,1}
wkj=net.LW{2,1}
wk0=net.b{2,1}

```

```

>> xi=-0.1061

>> z1=(1-exp(znet1))/(1+exp(znet1))
>> z2=(1-exp(znet2))/(1+exp(znet2))
>> z3=(1-exp(znet3))/(1+exp(znet3))
>> z4=(1-exp(znet4))/(1+exp(znet4))
>> z5=(1-exp(znet5))/(1+exp(znet5))
>> z6=(1-exp(znet6))/(1+exp(znet6))
>> z7=(1-exp(znet7))/(1+exp(znet7))
>> z8=(1-exp(znet8))/(1+exp(znet8))
>> z9=(1-exp(znet9))/(1+exp(znet9))

>> A1=(wkj(:,1)*z1)
>> A2=(wkj(:,2)*z2)
>> A3=(wkj(:,3)*z3)
>> A4=(wkj(:,4)*z4)
>> A5=(wkj(:,5)*z5)
>> A6=(wkj(:,6)*z6)
>> A7=(wkj(:,7)*z7)
>> A8=(wkj(:,8)*z8)
>> A9=(wkj(:,9)*z9)

>> A10=A1+A2+A3+A4+A5+A6+A7+A8+A9

>> ynet=wk0+A10

>> yk=ynet

>> hasilperamalan=poststd(yk,meant,stdt)

```